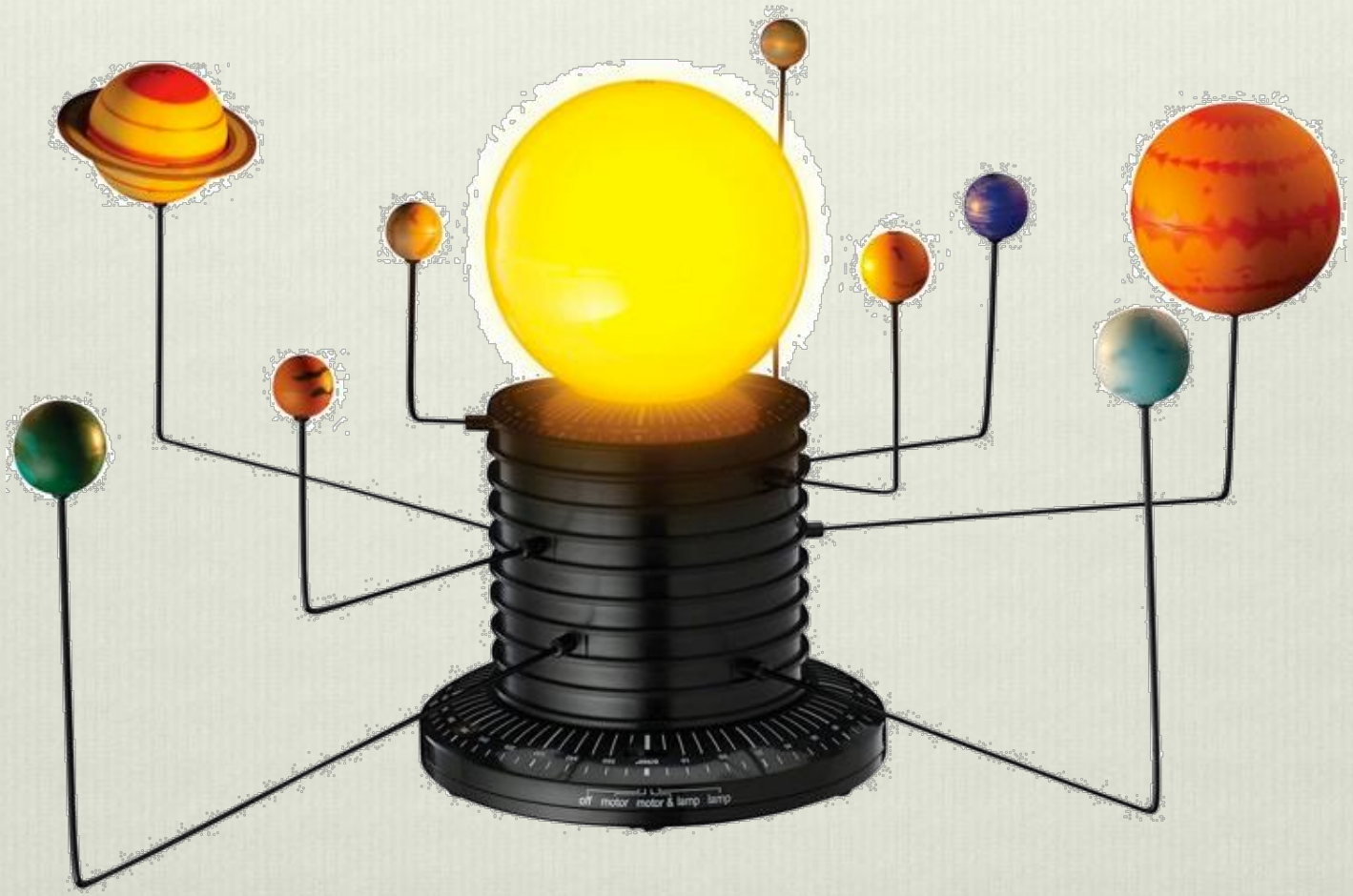
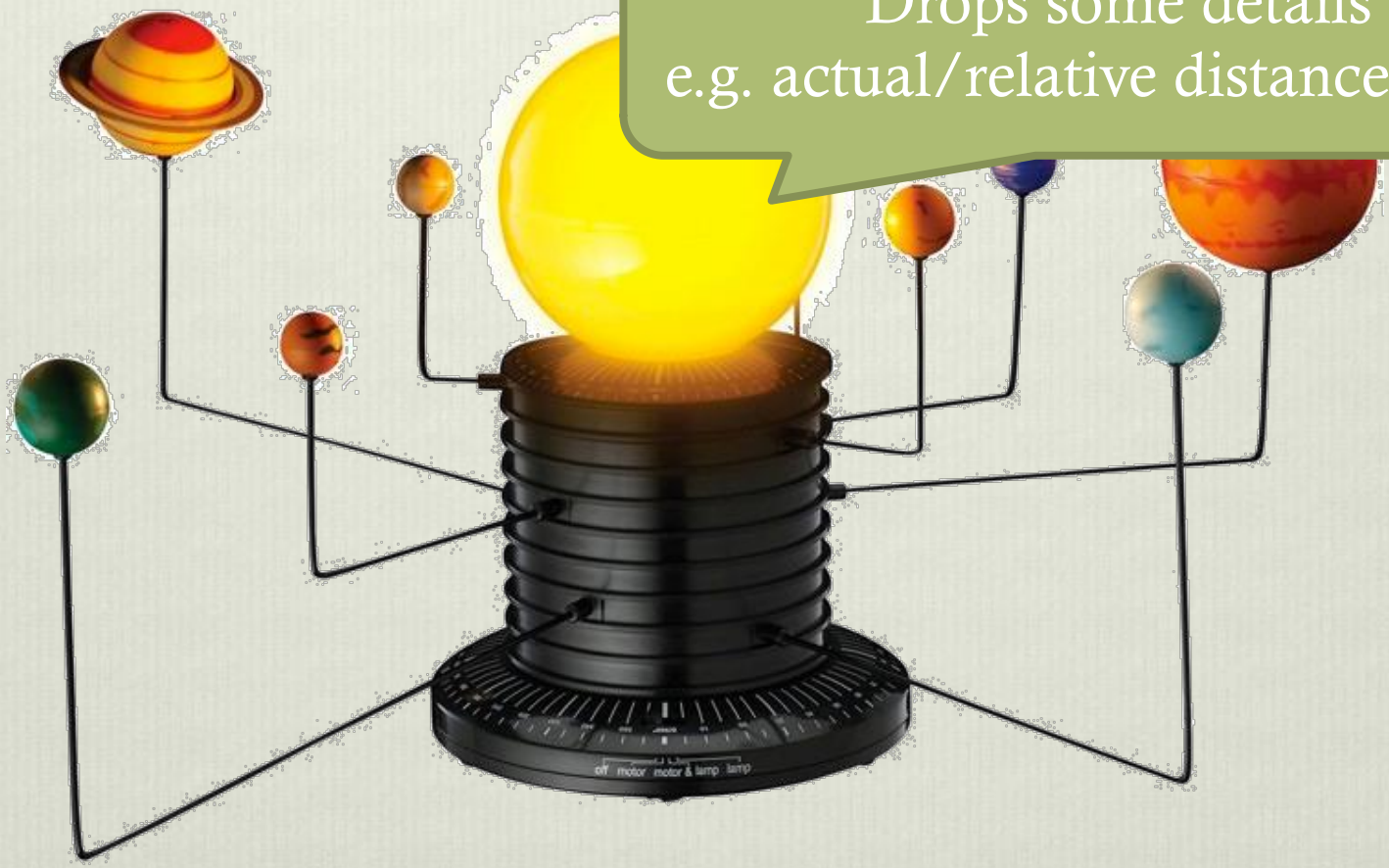
Model-based testing: What's in it for industry?

Christian Colombo

What is a model?

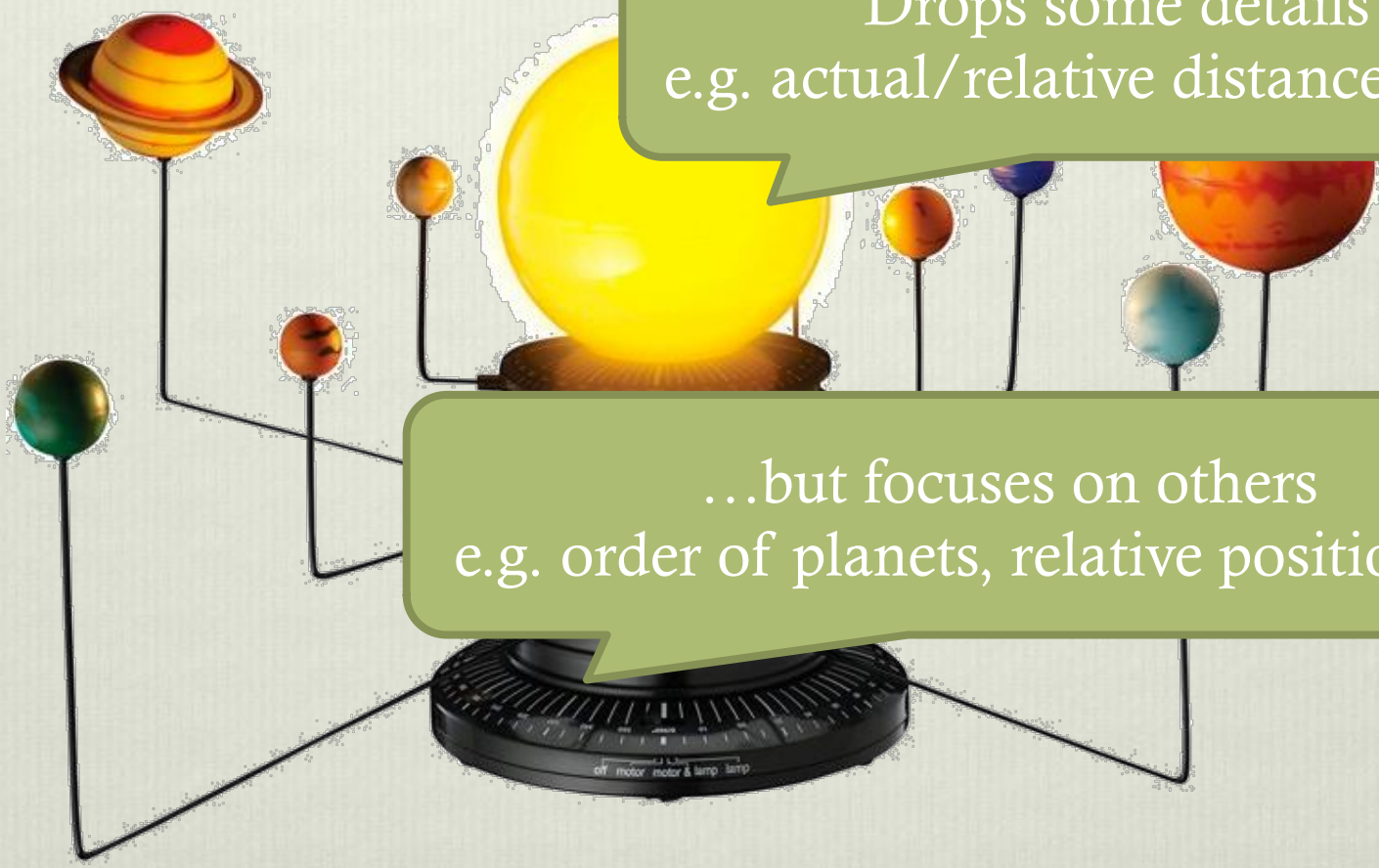


What is a model?



Drops some details
e.g. actual/relative distances/sizes

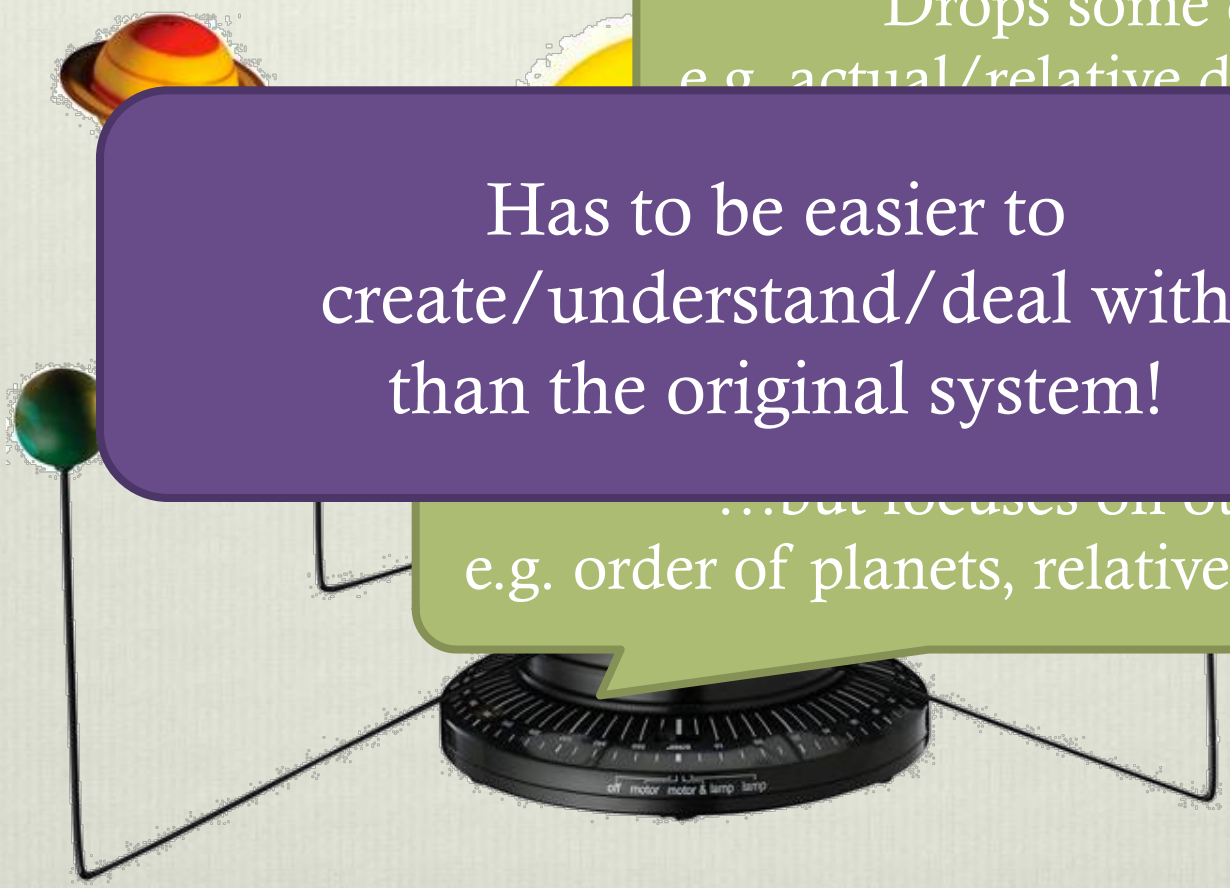
What is a model?



Drops some details
e.g. actual/relative distances/sizes

…but focuses on others
e.g. order of planets, relative positions, etc

What is a model?



Drops some details
e.g. actual/relative distances/sizes

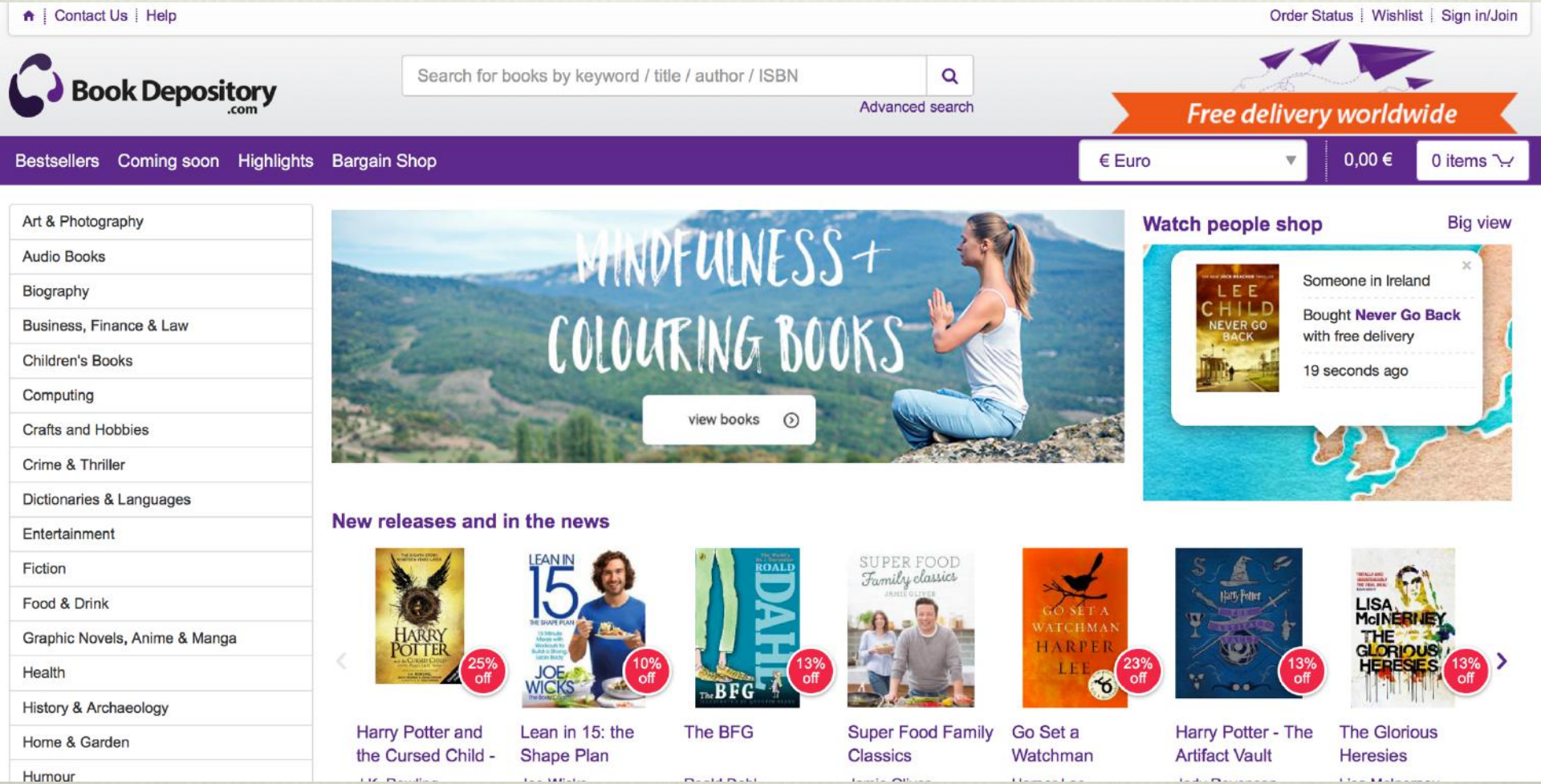
Has to be easier to
create/understand/deal with
than the original system!

…but focuses on others
e.g. order of planets, relative positions, etc

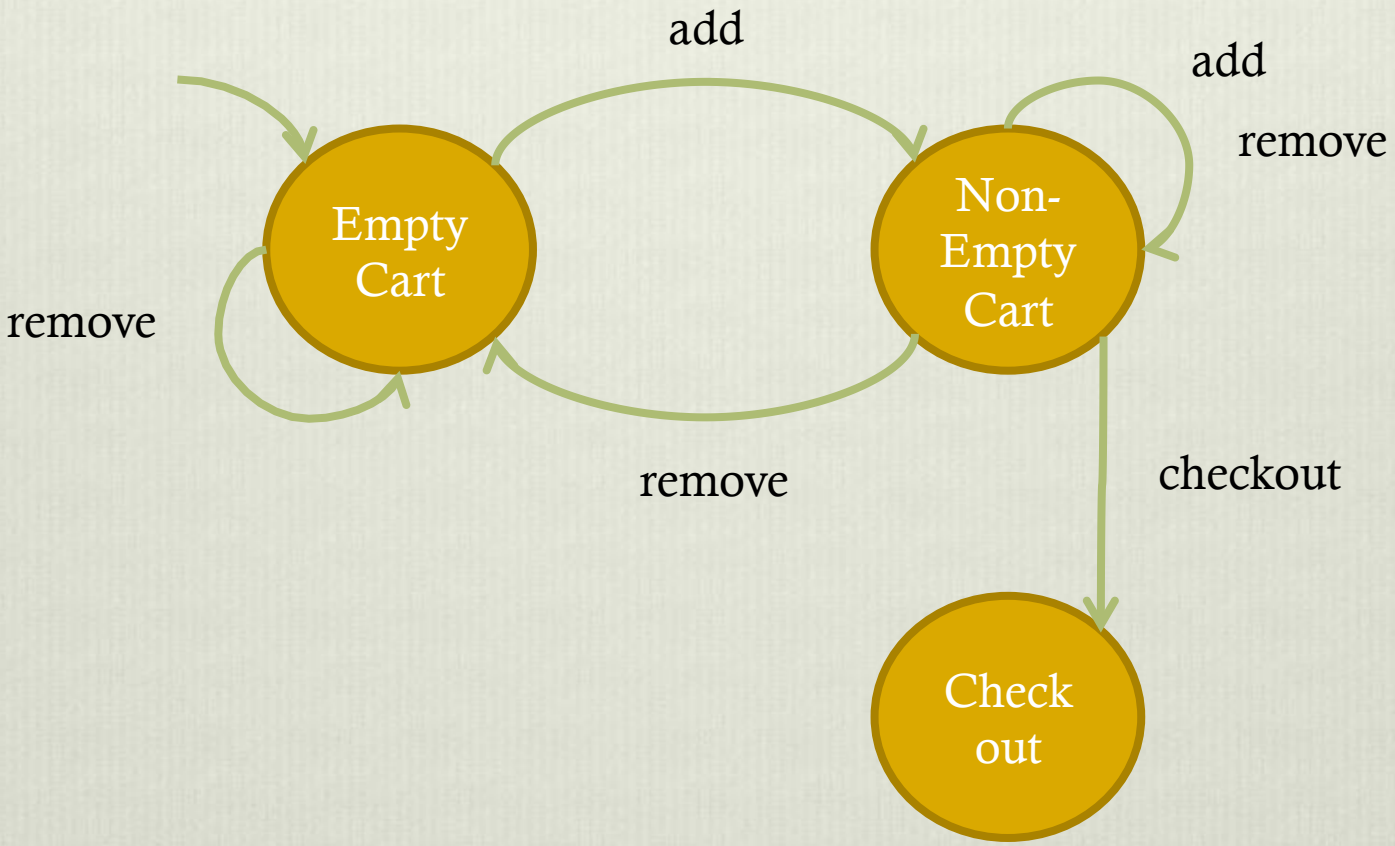
What to model in S/W?

- Expected order of actions
- Return values
- Timing

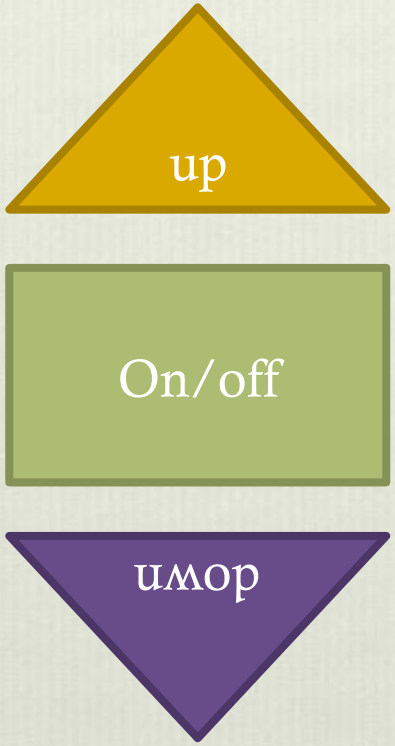
Example


Contact Us
Help
Order Status
Wishlist
Sign in/Join
Book Depository .com
Search for books by keyword / title / author / ISBN
Advanced search
Free delivery worldwide
Bestsellers
Coming soon
Highlights
Bargain Shop
€ Euro
0,00 €
0 items
Art & Photography
Audio Books
Biography
Business, Finance & Law
Children's Books
Computing
Crafts and Hobbies
Crime & Thriller
Dictionaries & Languages
Entertainment
Fiction
Food & Drink
Graphic Novels, Anime & Manga
Health
History & Archaeology
Home & Garden
Humour
MINDFULNESS + COLOURING BOOKS
view books
Watch people shop
Big view
Someone in Ireland
Bought Never Go Back with free delivery
19 seconds ago
New releases and in the news
25% off
Harry Potter and the Cursed Child -
10% off
Lean in 15: the Shape Plan
13% off
The BFG
Super Food Family Classics
23% off
Go Set a Watchman
13% off
Harry Potter - The Artifact Vault
13% off
The Glorious Heresies

Example


add
add
remove
Empty Cart
Non-Empty Cart
remove
remove
checkout
Check out

Light switch example



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

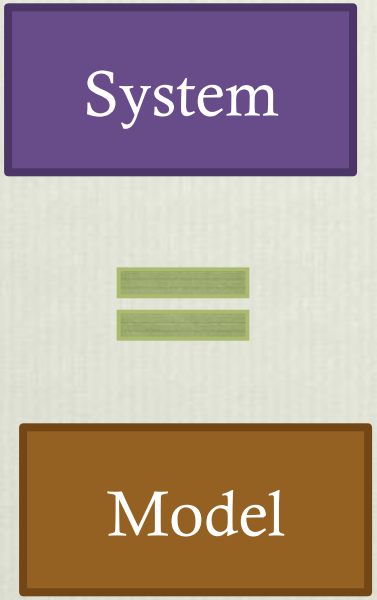
MBT can automate them all!

Test automation

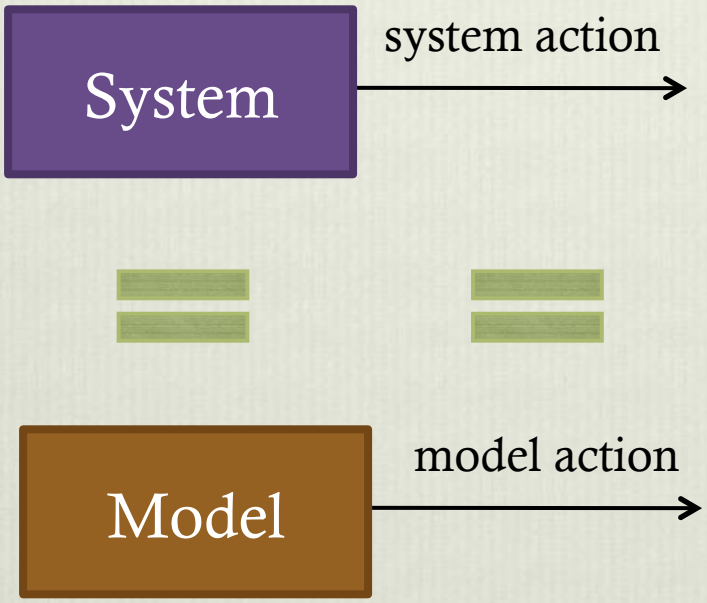
- Testcase generation
- Testcase execution
- Testcase pass/fail

MBT can automate them all!
(once you have the model)

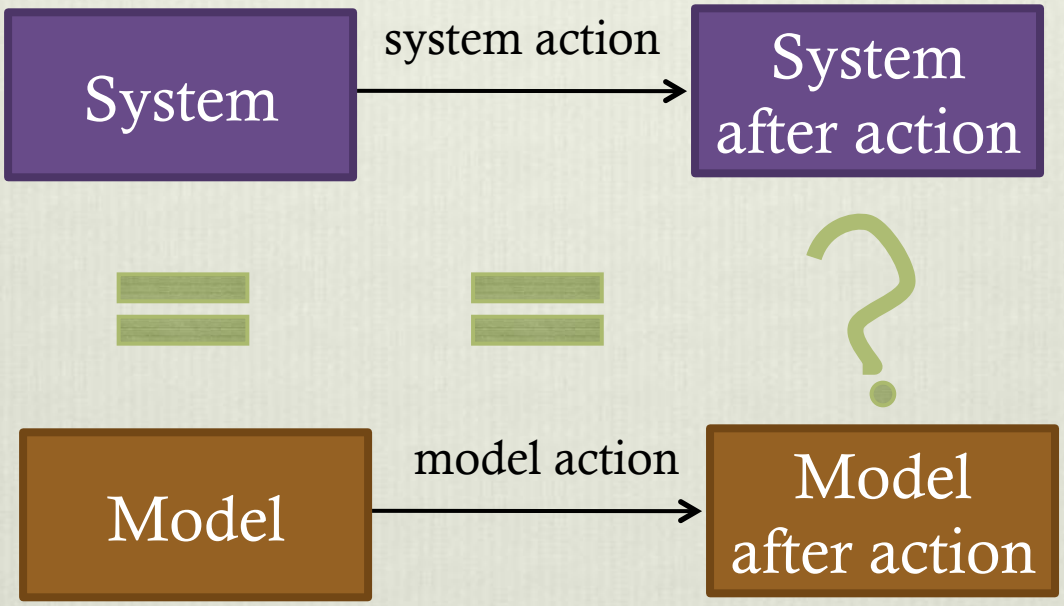
Model-Based Testing



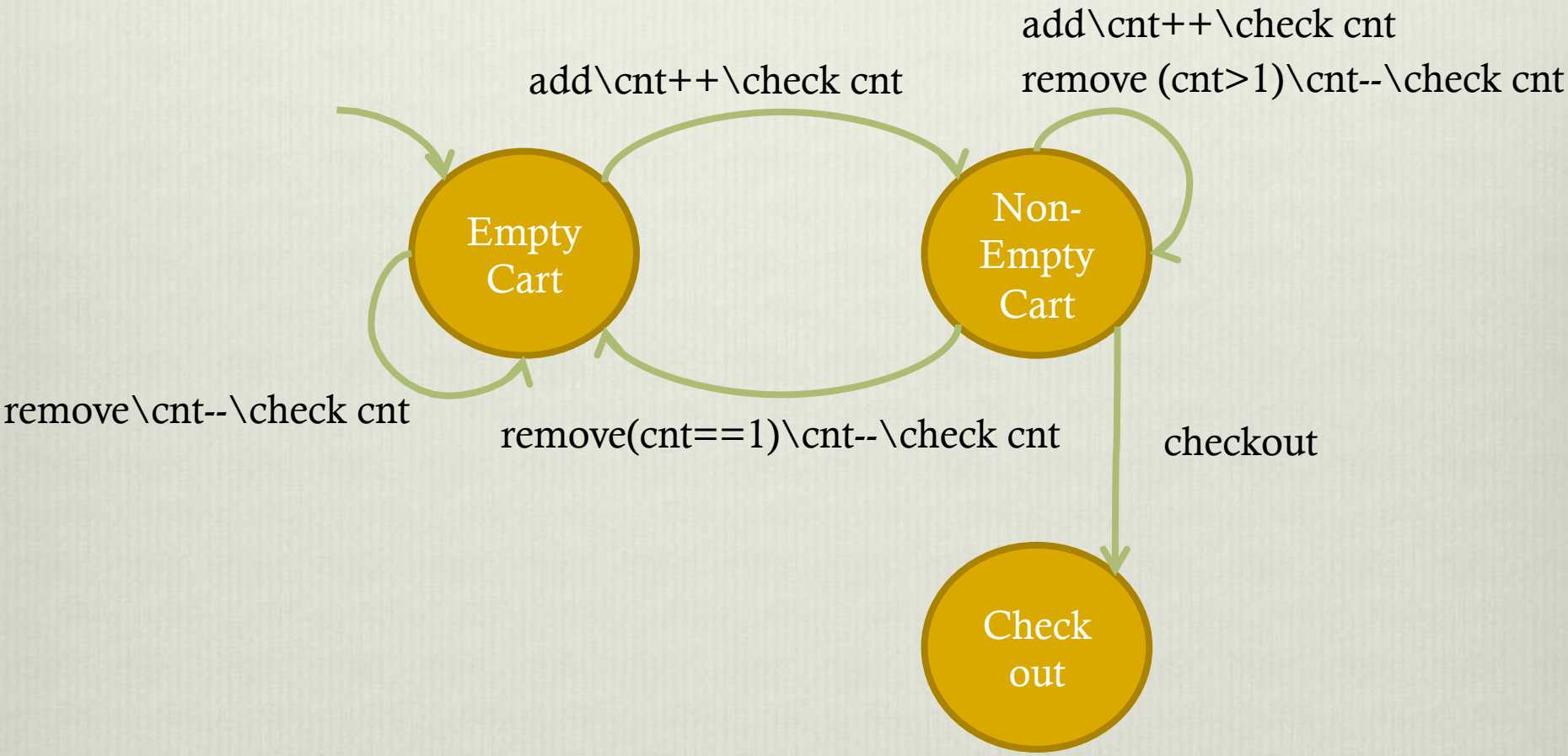
Model-Based Testing



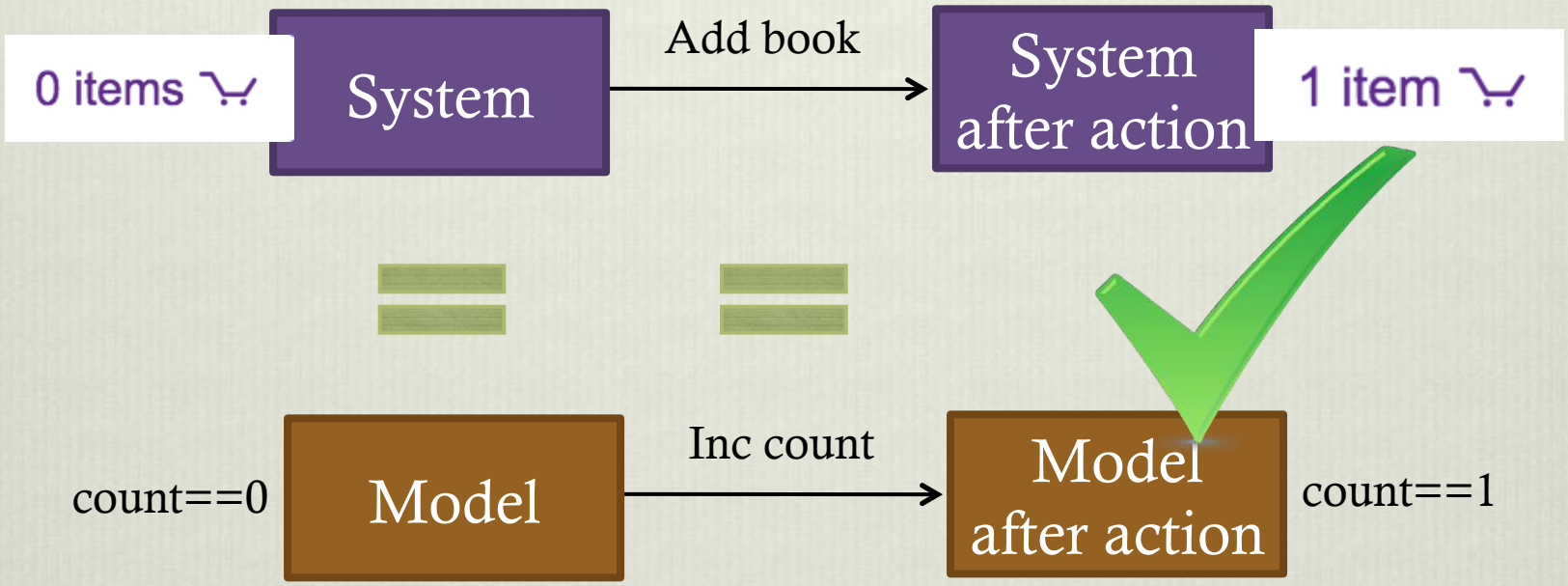
Model-Based Testing



Example

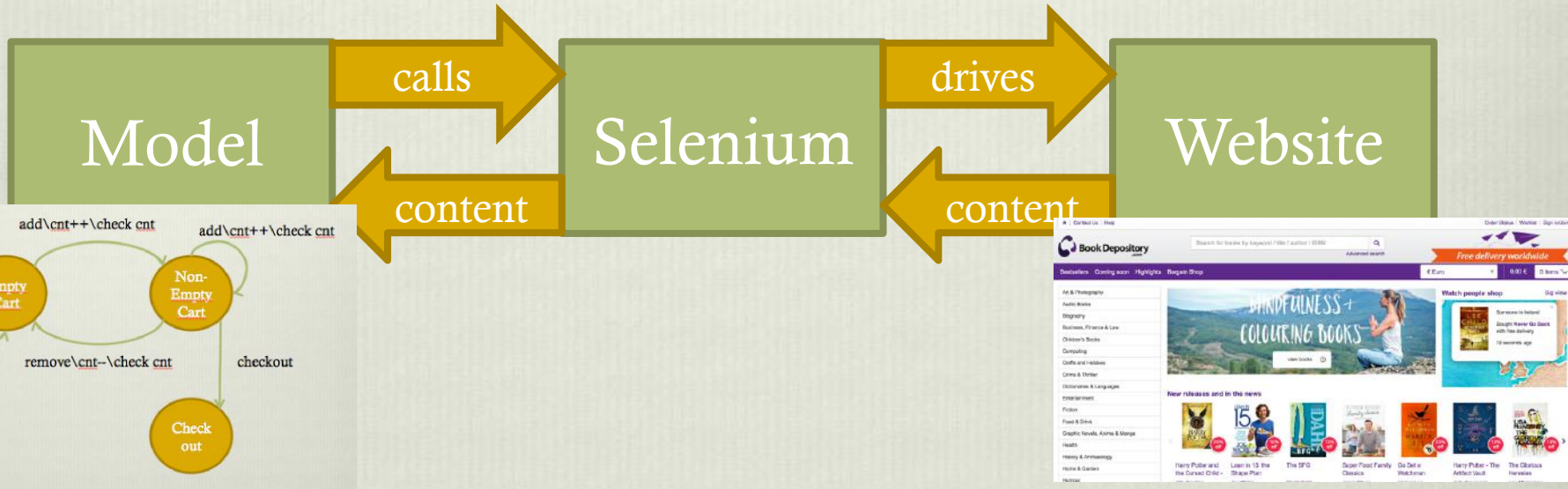

add\cnt++\check cnt
add\cnt++\check cnt
remove (cnt>1)\cnt--\check cnt
Empty Cart
Non-Empty Cart
remove\cnt--\check cnt
remove(cnt==1)\cnt--\check cnt
checkout
Check out

Model-Based Testing

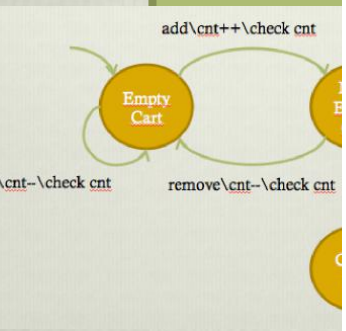


UI example

UI example

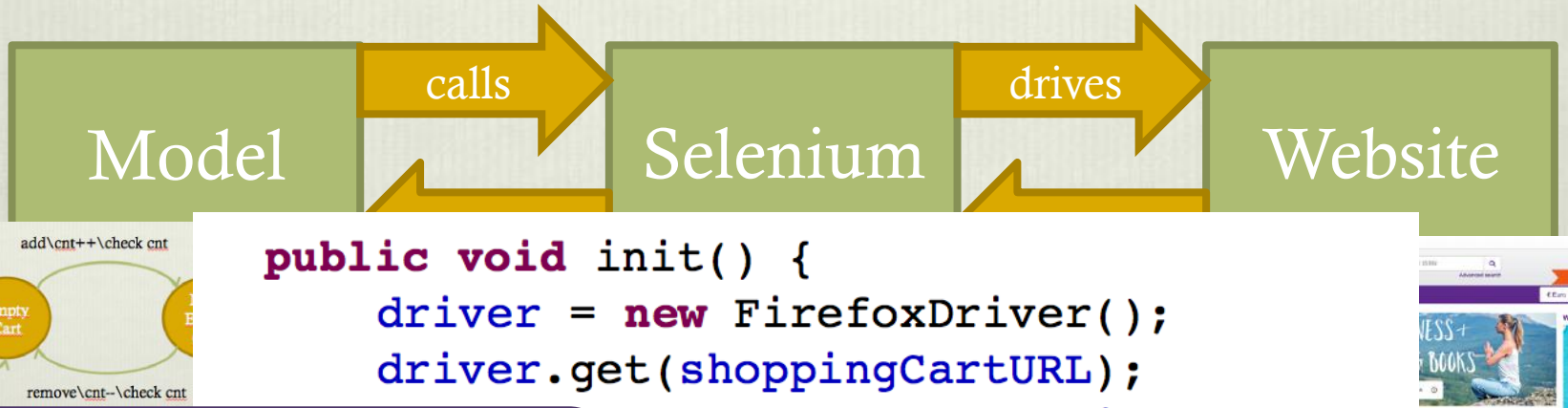

Model
calls
content
Selenium
drives
content
Website
add\cnt++\check cnt
add\cnt++\check cnt
Empty Cart
Non-Empty Cart
\cnt--\check cnt
remove\cnt--\check cnt
checkout
Check out
Book Depository
Free delivery worldwide
MINDFULNESS + COLOURING BOOKS

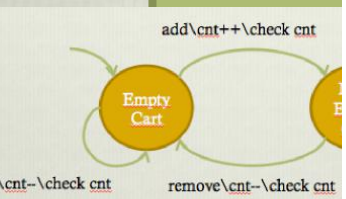
UI example




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

public void addItem() {
    searchPage.searchAndAddToCart(null);
}
```

UI example






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

    addItem() {
    searchPage.searchAndAddToCart(null);
}
```

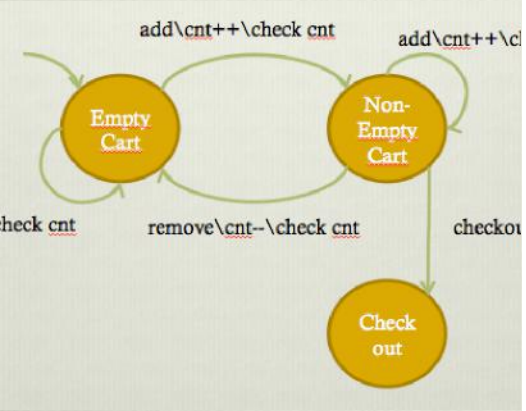
How do you code the model?

Tools for MBT

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

http://mit.bme.hu/~micskeiz/pages/modelbased_testing.html

ModelJUnit - States

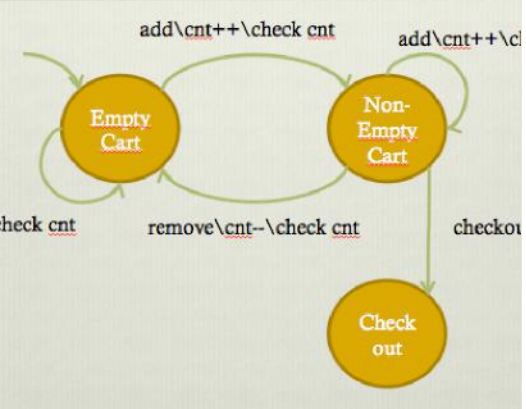



```
//States
public enum WebsiteState {
    EMPTY_CART,
    NON_EMPTY_CART,
    CHECKOUT,
    ERROR_STATE
}

// 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;
}
```

The state names

Variables

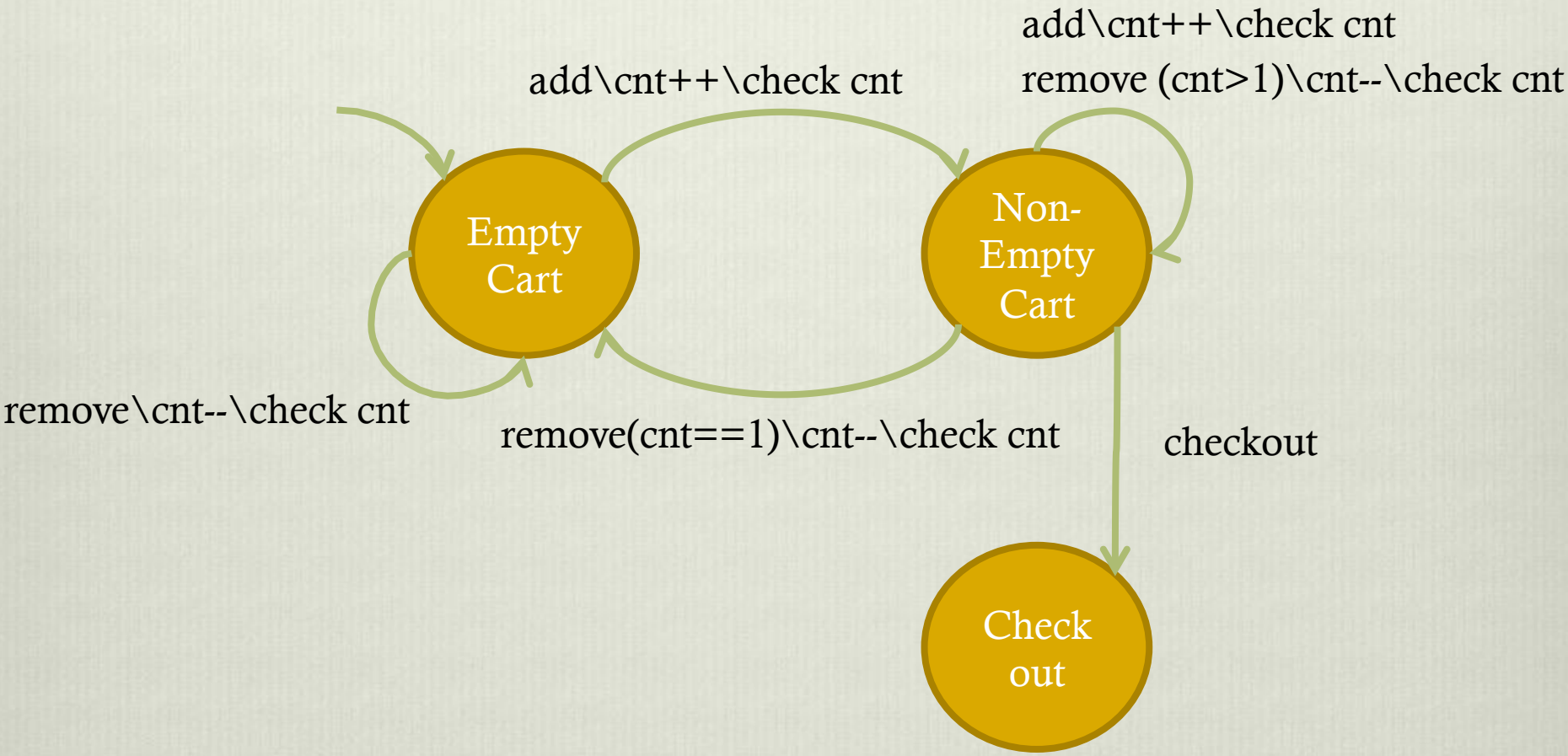
ModelJUnit - States




```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


add\cnt++\check cnt
add\cnt++\check cnt
remove (cnt>1)\cnt--\check cnt
Empty Cart
Non-Empty Cart
remove\cnt--\check cnt
remove(cnt==1)\cnt--\check cnt
checkout
Check out

ModelJUnit - Transitions

```
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

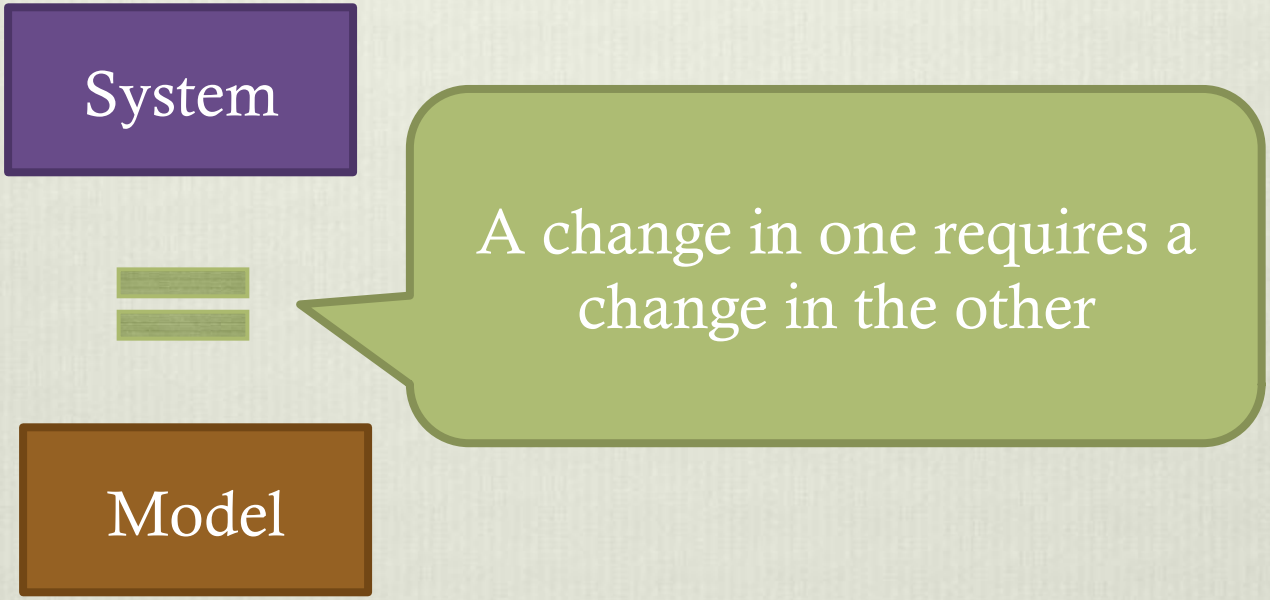
Benefits of MBT

- Automatic generation of test cases
- Automatic verification of tests

Challenges of MBT

- Writing the model
- Some learning curve

Maintaining the correspondence



The End

- Questions