



Tutorial MBT for Beginners



Eddie Jaffuel
Consultant

Agenda

- Introduction (15 min)
- MBT as a black box (35 min)
- Open the box (20 min)
- The value of MBT (10 min)
- Questions (10 min)



Agenda

- Introduction
 - The speaker
 - What is MBT ?
 - Goal of the tutorial
 - Focus of the tutorial
 - MBT Process overview
- MBT as a black box
- Open the box
- The value of MBT
- Questions



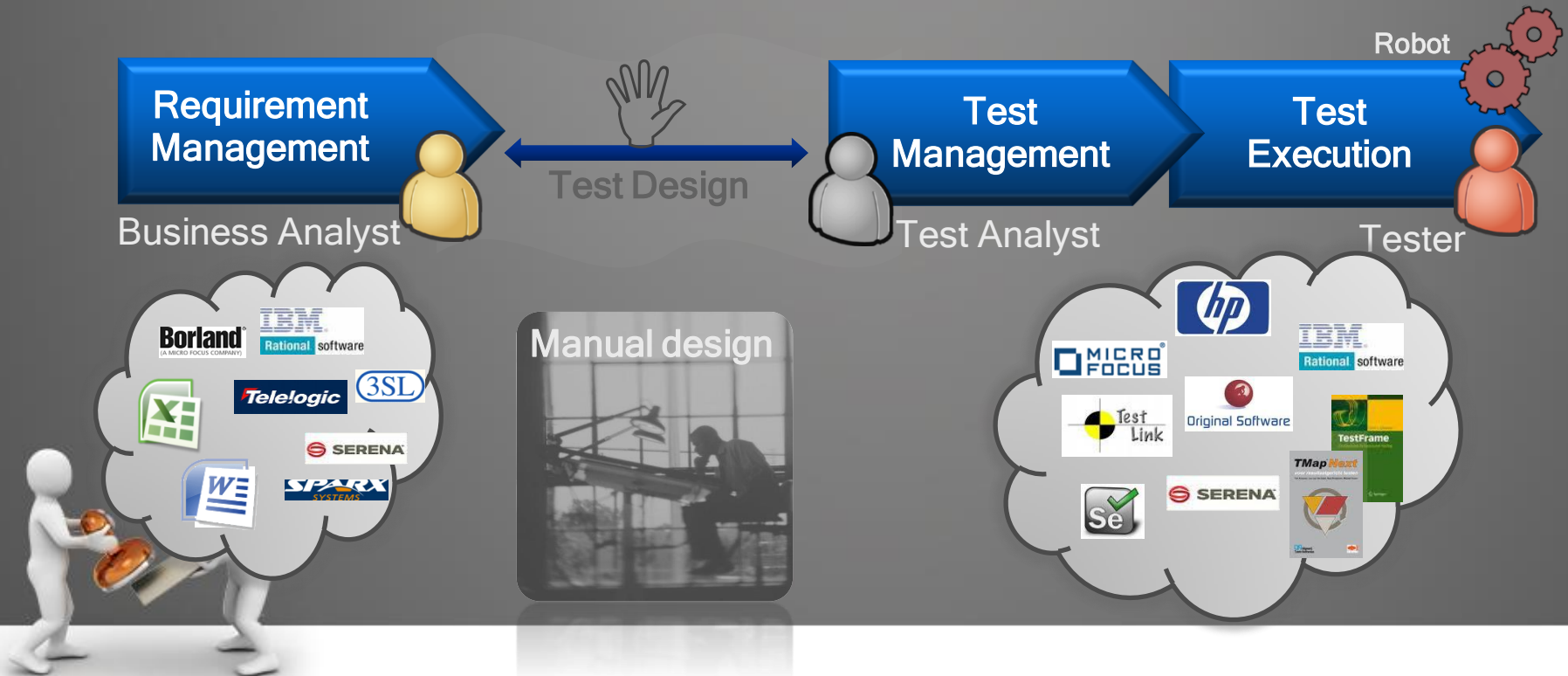
Who is Eddie JAFFUEL?

- Independent consultant
- Expert in Modeling and Model Based Testing for 10 years
 - Proof Of Concepts
 - Deployment Pilot Projects
 - Training, Coaching and Knowledge Transfer
 - Model Production
 - Technical Lead and Project Manager for the MBT technics:
 - Information Technologies
 - Automotive industries
 - Embedded Systems (Smartcard & Device industries)
- MBT applied to **GLOBALPLATFORM™** Compliance Testing (KeyNote)
- More information... **Linked in**



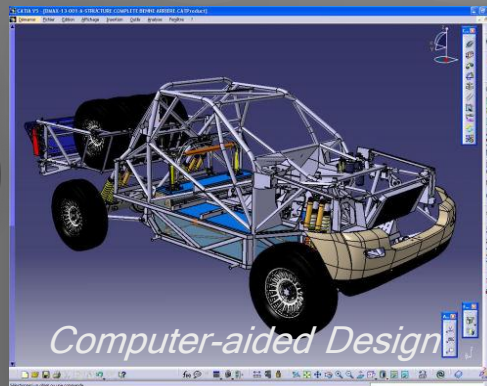
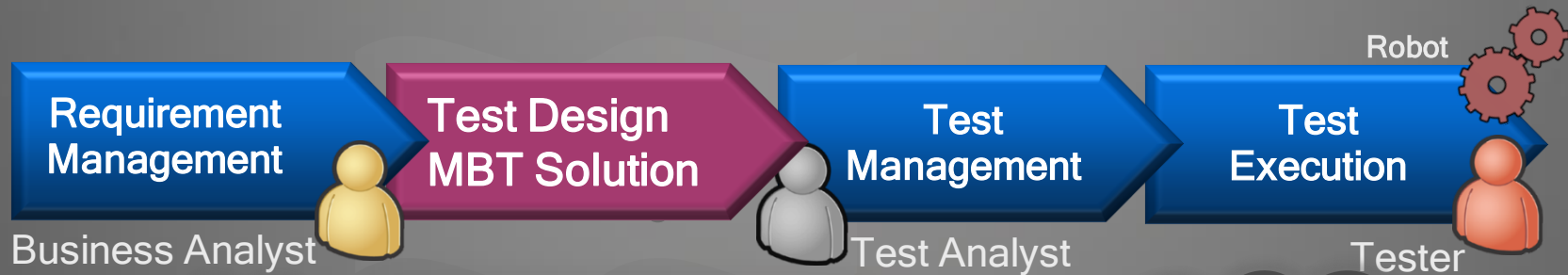
What is MBT ?

- The context:



What is MBT ?

- MBT = Model Based Testing
- Focused on Test Design
- Combines a method: Model-Based
- And a Technology: automatic test generation from a model



Goal of the tutorial

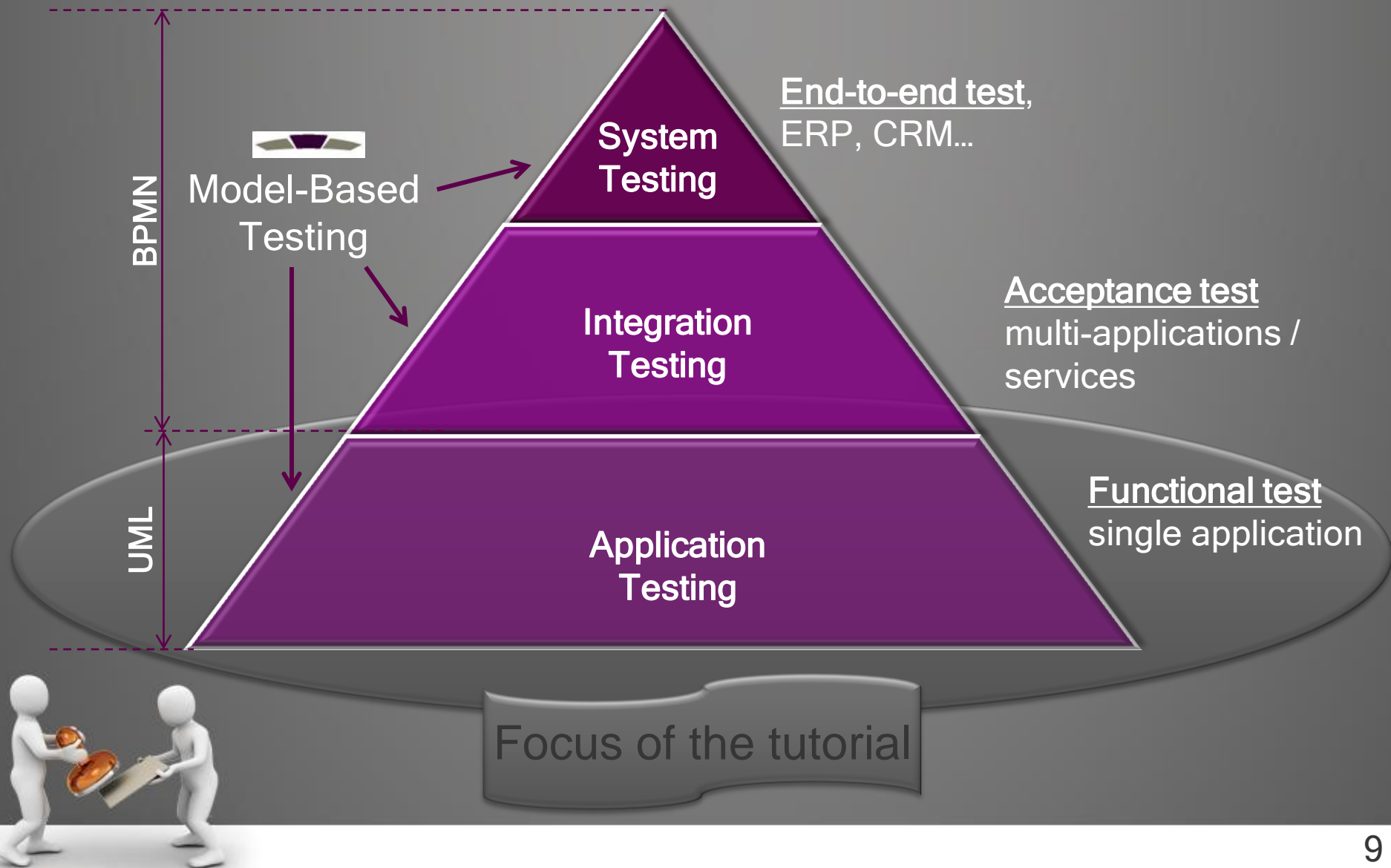
- Understand the principles of MBT
 - How a MBT Tool may deduce tests from a Model ?
- Explanation based on One MBT Methodology
 - from the Requirements...
 - through the Test Objective Charter...
 - through the Behavioral Model
 - to the generated Tests for manual execution
- Benefits of MBT approach
 - Different levels of factorization,
 - Ability to manage changes more efficiently through MBT



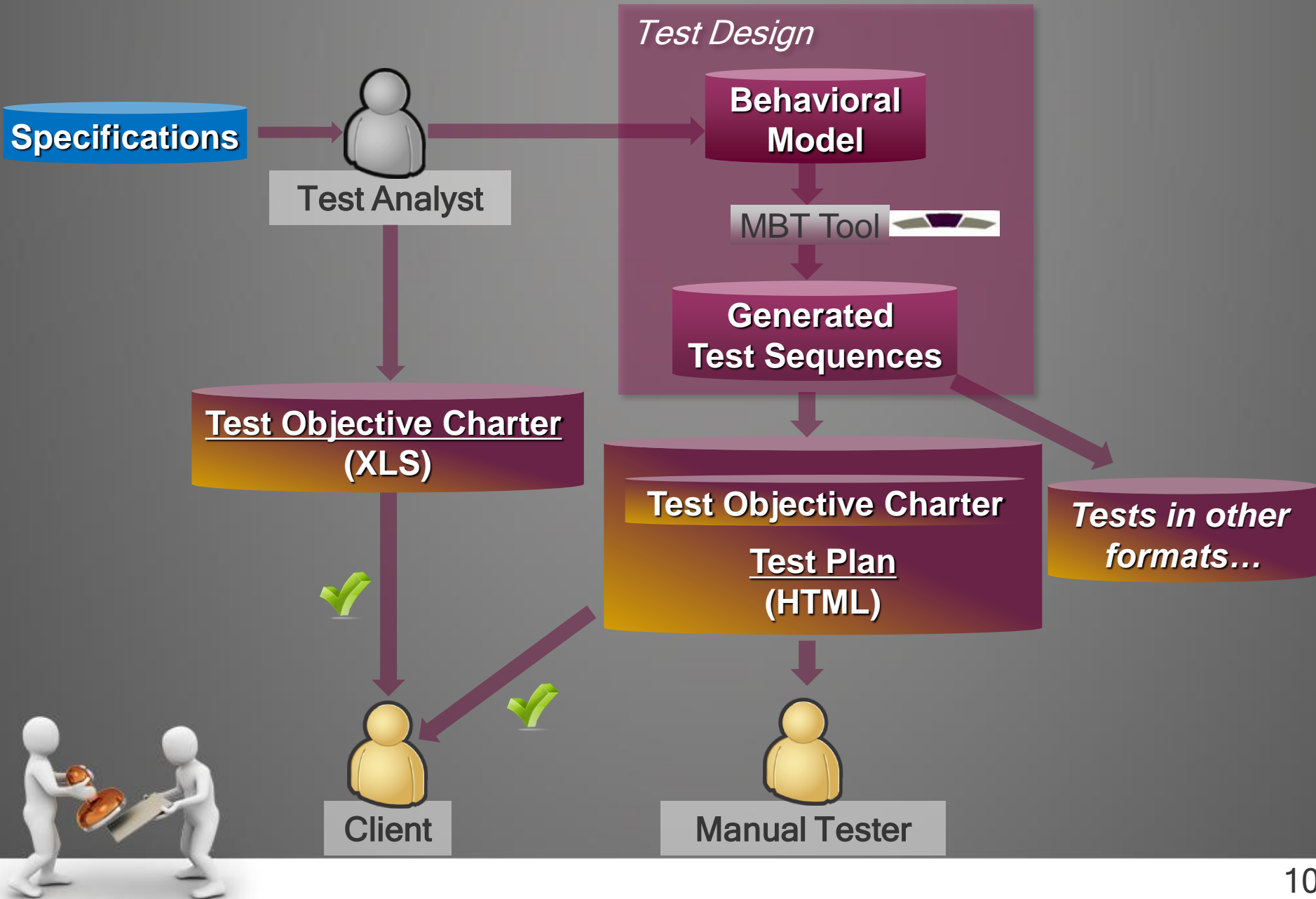
Focus of the Tutorial

<u>Criteria</u>	<u>Focus of the tutorial</u>	<u>Not covered by tutorial but MBT is also applicable for</u>
Kind of System Under Test	Web based application	Application Java / ERP / CRM Multi-Applications System / ...
Interface to access to SUT	Graphical User Interface	API / Services / SOA Flow / ...
Level of test	Application Testing Functional Testing	System Testing Integration Testing Compliance Testing ...
Nature of test execution	Manual Execution	Automated Execution
Test format	HTML	TTCN, XML, script languages, VB, C, Java ...
Model	UML (Unified Modeling Language)	BPMN, ...
MBT Solution/Product	Smartesting CertifyIt	Conformiq, Matelo, SpecExplorer,

What kind of tests ?



MBT Process Overview



Agenda

- Introduction
- **MBT as a black box**
 - Introduction of the System Under Test
 - Test Objective Charter
 - Behavioral Model
 - Test Plan Publication
- Open the box
- The value of MBT
- Questions



Introduction of the System Under Test (1/4)

- It is a simple Web Application : Sims
- Features
 - Manage the inhabitants (Add / Delete / Edit)
 - Marry the inhabitants
 - Divorce the inhabitants
- Navigation
 - Start at HOME
 - Go to CHURCH to perform a Marriage, once done go back HOME
 - Go to COURT to perform a Divorce, once done: go back HOME



Introduction of the System Under Test (2/4)

	First name	Last name	Gender	Age	Married with	Edit	Delete
	Isabel	H	Female	34	--		
	Eddie	Jaffuel	Male	39	--		
	Dooley	NSEWOLO	Male	35	Christelle NSEWOLO		
	Christelle	NSEWOLO	Female	30	Dooley NSEWOLO		

New Inhabitant Go to church Go to Court

Firefox

Sims

First name

Last name

Gender

Male

Age





Create Inhabitant

Introduction of the System Under Test (3/4)

Firefox

Sims +

Smart-Sims in Church

	First name	Last name	Gender	Age	Married with
	Isabel	H	Female	34	--
	Eddie	Jaffuel	Male	39	--
	Dooley	NSEWOLO	Male	35	Christelle NSEWOLO
	Christelle	NSEWOLO	Female	30	Dooley NSEWOLO

Inhabitants list **Get married**

New marriage

First person
Isabel H

Second person
Eddie Jaffuel

Get Married

New marriage

1 error prohibited this wedding from being saved:

- Second spouse already married

First person
Isabel H

Second person
Dooley NSEWOLO

Get Married

Introduction of the System Under Test (4/4)

First name	Last name	Gender	Age	Married with
Isabel	H	Female	34	Eddie Jaffuel
Eddie	Jaffuel	Male	39	Isabel H
Dooley	NSEWOLO	Male	35	Christelle NSEWOLO
Christelle	NSEWOLO	Female	30	Dooley NSEWOLO

Inhabitants list **Divorce**

New divorce

First person
Isabel H

Second person
Eddie Jaffuel

Divorce

Agenda

- Introduction
- MBT as a black box
 - Introduction of the System Under Test
 - Test Objective Charter **Test Objective Charter**
 - Behavioral Model
 - Test Plan Publication
- Open the box
- The value of MBT
- Questions



- a « Test Objective Charter » (TOC) helps to
 - List and Identify the REQuirements or Business Rules to cover
 - Describe the Expected Coverage (Test AIMS)

➔ WHAT to cover



Test Objective Charter (2/3)

@REQ	Requirement description	#AIM			
ADD_INHABITANT	<p>The conditions which allow to add one inhabitant are:</p> <ul style="list-style-type: none">- all its informations are provided (identifier, gender, age)- its age have to be greated than 1- the inhabitant not already exist <p>In case of success, the inhabitant is added to the list of inhabitants, and its information are displayed. A new inhabitant is single by default.</p> <p>If an error occurs, an error message indicates which condition is not fullfilled.</p>	Success	Already exists	Empty identifier	Age not strictly positive
DELETE_INHABITANT	You can only suppress one inhabitant if it exists	Success			

Test Objective Charter (3/3)

@REQ	Requirement description	#AIM				
MARRIAGE	<p>The conditions which allow a marriage are:</p> <ul style="list-style-type: none"> - age over 18 - none of them are married - one male and one female <p>Once the marriage is accepted, the status of the 2 inhabitants is modified accordingly their fields "Married with ..." is filled</p> <p>If an error occurs, an error message indicates which condition is not fullfilled.</p>	Success	Error same person	Error same gender	Error one is not adult	Error one is already married



Agenda

- Introduction
- **MBT as a black box**
 - Introduction of the System Under Test
 - Test Objective Charter
 - **Behavioral Model**
 - Test Plan Publication
- Open the box
- The value of MBT
- Questions

**Behavioral
Model**



- The different components of the Behavioral Model:
 - Static part
 - Points of Control and Observation → Operations **Point Control Obs**
 - Data Representation → Classes
 - Dynamic part
 - Initial Data of the System Under Test (SUT) → Objects
 - Flows → State Machine
 - Business Rules → OCL: Object Constraint Language



Behavioral Model: Points of Control → Operations

Behavioral
Model

Point Control Obs


Sims_app

- Add_inhabitant ()
- Go_to_church ()
- Go_to_court ()
- Delete_inhabitant ()

Firefox

Sims

Smart-Sims at Home

	First name	Last name	Gender	Age	Married with	Edit	Delete
	Isabel	H	Female	34	--		
	Eddie	Jaffuel	Male	39	--		
	Dooley	NSEWOLO	Male	35	Christelle NSEWOLO		
	Christelle	NSEWOLO	Female	30	Dooley NSEWOLO		

New Inhabitant Go to church Go to Court



Behavioral Model: Points of Control → Operations + Parameters

Behavioral
Model

Point Control Obs

Sims_app

Add_inhabitant ()

- IN_id
- IN_gender
- IN_age

Firefox

Sims

First name

Last name

Gender

Male

Age

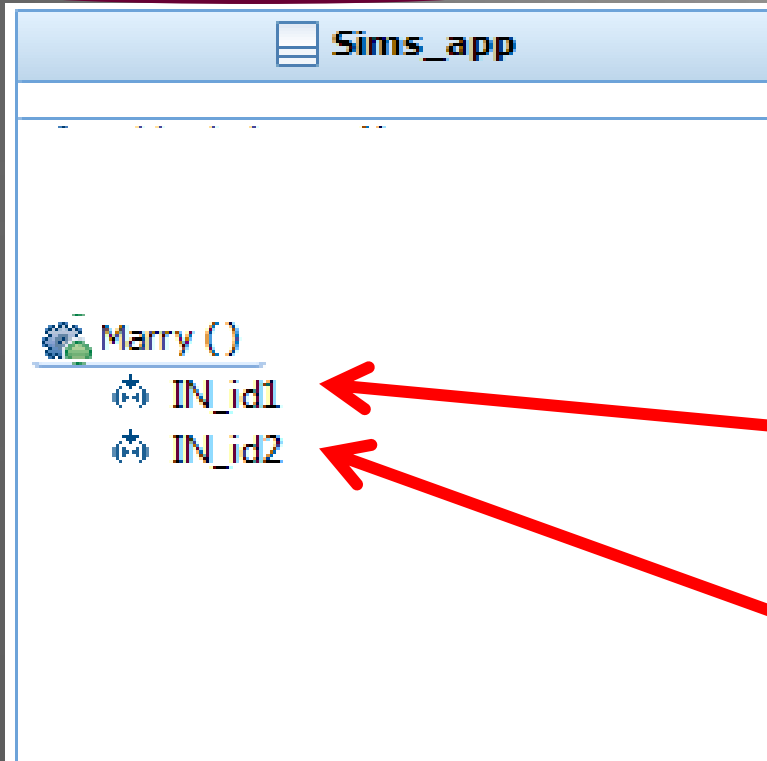
Create Inhabitant



Behavioral Model: Points of Control → Operations + Parameters

Behavioral
Model

Point Control Obs



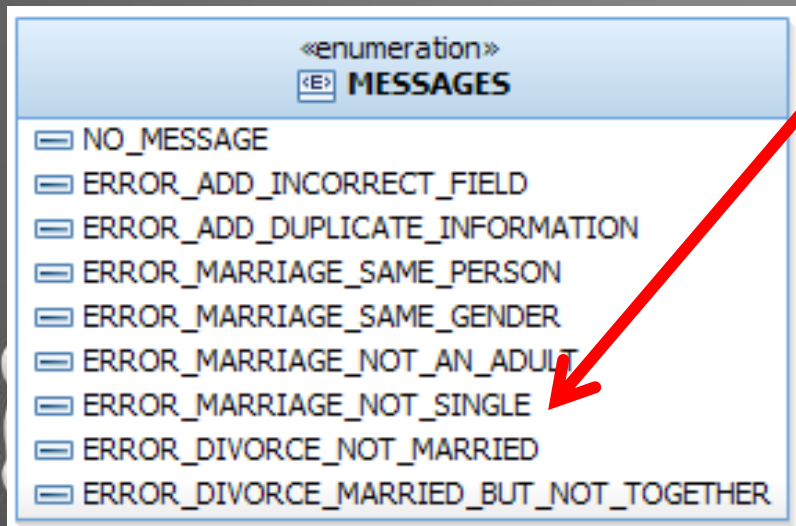
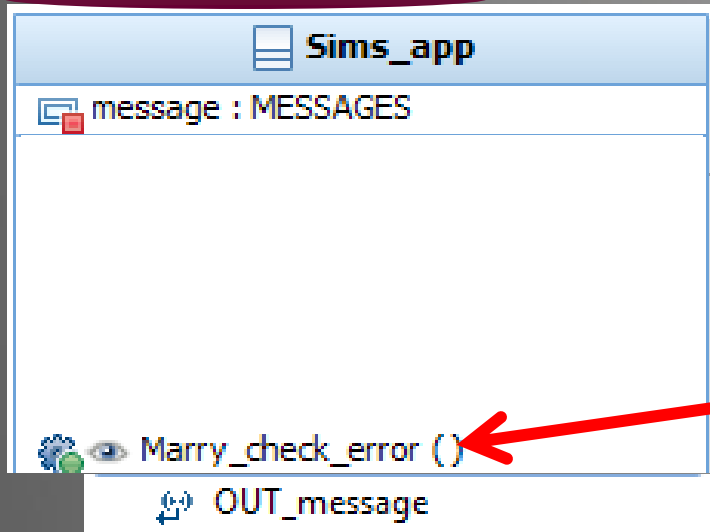
A screenshot of a 'New marriage' form. The form has a title 'New marriage' in a red header. Below the title, there are two dropdown menus: 'First person' with the value 'Isabel H' and 'Second person' with the value 'Eddie Jaffuel'. At the bottom of the form is a 'Get Married' button. Red arrows point from the 'Isabel H' and 'Eddie Jaffuel' dropdowns to the 'IN_id1' and 'IN_id2' parameters in the Sims application window on the left.



Behavioral Model: Points of Observation → Operations

Behavioral
Model

Point Control Obs



New marriage

1 error prohibited this wedding from being saved:

- Second spouse already married

First person
Isabel H

Second person
Dooley NSEWOLO

Get Married

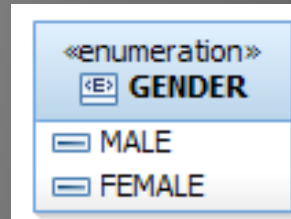
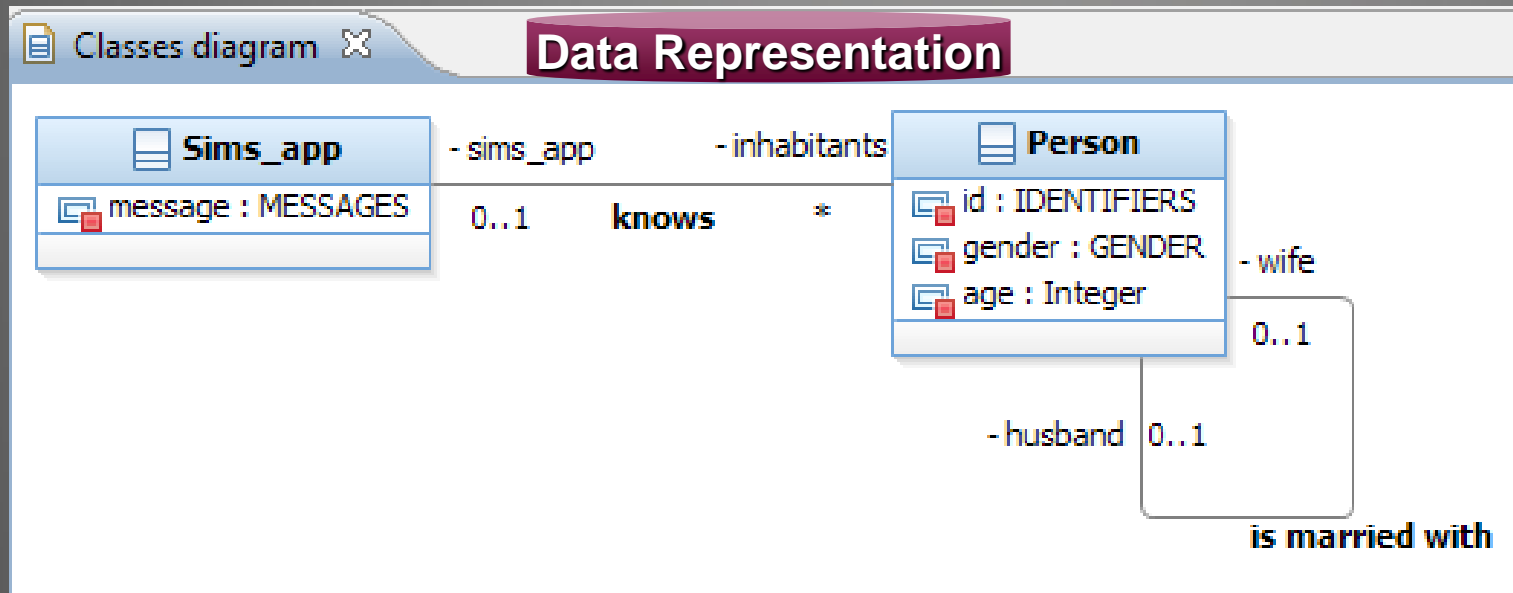
Behavioral Model

- The different components of the Behavioral Model:
 - Static part
 - Points of Control and Observation → Operations
 - Data Representation → Classes **Data Representation**
 - Dynamic part
 - Initial Data of the System Under Test (SUT) → Objects
 - Flows → State Machine
 - Business Rules → OCL: Object Constraint Language



Behavioral Model : Data Representation

- The **Class diagrams** helps to model the Data representation
 - **Classes** helps to represent the Business Entities
 - **Attributes** are the characteristics of the Business Entities
 - **Association** are the relations between Business Entities



- The different components of the Behavioral Model:
 - Static part
 - Points of Control and Observation → Operations
 - Data Representation → Classes
 - Dynamic part
 - Initial Data of the System Under Test (SUT) → Objects **Initial Data of SUT**
 - Flows → State Machine
 - Business Rules → OCL: Object Constraint Language







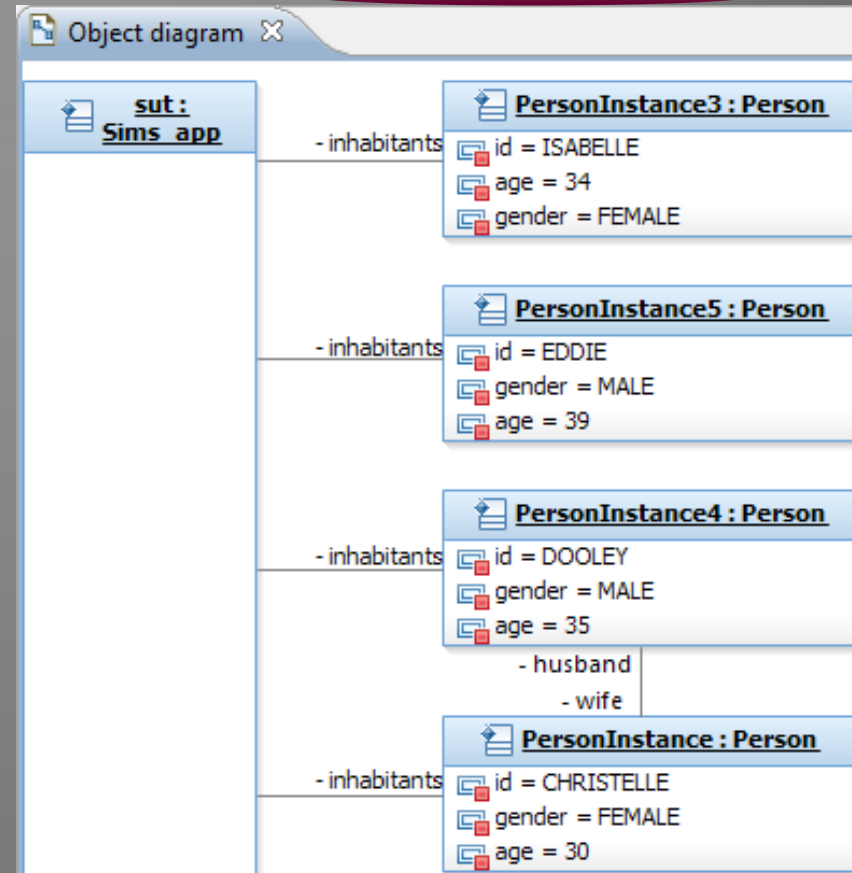
Behavioral Model: Initial Data of the SUT

Behavioral
Model

- The Object diagram used to model the Initial Data of the SUT
 - Objects are instance of Business Entities

Initial Data of SUT

	First name	Last name	Gender	Age	Married with
	Isabel	H	Female	34	--
	Eddie	Jaffuel	Male	39	--
	Dooley	NSEWOLO	Male	35	Christelle NSEWOLO
	Christelle	NSEWOLO	Female	30	Dooley NSEWOLO

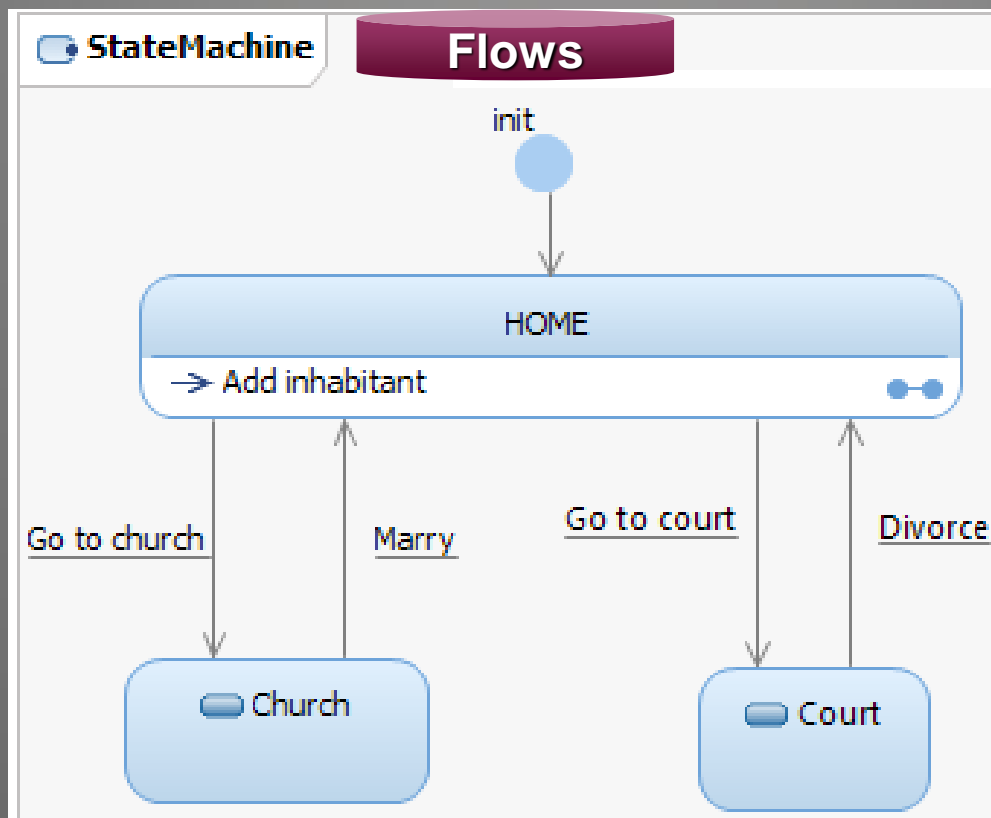


- The different components of the Behavioral Model:
 - Static part
 - Points of Control and Observation → Operations
 - Data Representation → Classes
 - Dynamic part
 - Initial Data of the System Under Test (SUT) → Objects
 - Flows → State Machine **Flows**
 - Business Rules → OCL: Object Constraint Language



Behavioral Model: Dynamic Flows

- The **State Machine** used to model the Dynamic Flows
 - With States and Transitions



- The different components of the Behavioral Model:
 - Static part
 - Points of Control and Observation → Operations
 - Data Representation → Classes
 - Dynamic part
 - Initial Data of the System Under Test (SUT) → Objects
 - Flows → State Machine
 - Business Rules → OCL: Object Constraint Language

Business Rules



Behavioral Model: Business Rules

Test Objective Charter

@REQ	MARRIAGE
Requirement description	<p>The conditions which allow a marriage are:</p> <ul style="list-style-type: none">- age over 18- none of them are married- one male and one female <p>Once the marriage is accepted, the status of the 2 inhabitants is modified accordingly their fields "Married with ..." is filled</p> <p>If an error occurs, an error message indicates which condition is not fulfilled.</p>
#AIM	Error same person
	Error same gender
	Error one is not adult
	Error one is already married
	Success

Business Rules

```
*Marry ✕  
---@REQ: MARRIAGE  
if person1 = person2 then  
    ---@AIM: Error same person  
    self.message = MESSAGES::ERROR_MARRIAGE_SAME_PERSON  
  
else if person1.gender = person2.gender then  
    ---@AIM: Error same gender  
    self.message = MESSAGES::ERROR_MARRIAGE_SAME_GENDER  
  
else if (person1.age < 18) or (person2.age < 18) then  
    ---@AIM: Error one is not adult  
    self.message = MESSAGES::ERROR_MARRIAGE_NOT_AN_ADULT  
  
else if person1.is_married() or person2.is_married() then  
    ---@AIM: Error one is already married  
    self.message = MESSAGES::ERROR_MARRIAGE_NOT_SINGLE  
  
else  
    ---@AIM: Success  
    self.message = MESSAGES::NO_MESSAGE  
    and  
    if person1.gender = GENDER::MALE then  
        ---@AIM: person1 is a man  
        person1.wife = person2  
    else  
        ---@AIM: person1 is a woman  
        person1.husband = person2  
    endif  
endif
```

Behavioral Model: Business Rules

Test Objective Charter

```

*Marry ✕
---@REQ: MARRIAGE
if person1 = person2 then
    ---@AIM: Error same person
    self.message = MESSAGES::ERROR_MARRIAGE_SAME_PERSON

else if person1.gender = person2.gender then
    ---@AIM: Error same gender
    self.message = MESSAGES::ERROR_MARRIAGE_SAME_GENDER

else if (person1.age < 18) or (person2.age < 18) then
    ---@AIM: Error one is not adult
    self.message = MESSAGES::ERROR_MARRIAGE_NOT_AN_ADULT

else if person1.is_married() or person2.is_married() then
    ---@AIM: Error one is already married
    self.message = MESSAGES::ERROR_MARRIAGE_NOT_SINGLE

else
    ---@AIM: Success
    self.message = MESSAGES::NO_MESSAGE
    and
    if person1.gender = GENDER::MALE then
        ---@AIM: person1 is a man
        person1.wife = person2
    else
        ---@AIM: person1 is a woman
        person1.husband = person2
    endif
endif
    
```

@REQ	MARRIAGE
Requirement description	<p>The conditions which all are:</p> <ul style="list-style-type: none"> - age over 18 - none of them are married - one male and one female <p>Once the marriage is accepted, the status of the 2 inhabitants is recorded accordingly their fields "is_married" is filled</p> <p>If an error occurs, an error message indicates which condition is not fulfilled.</p>
#AIM	Error same person
	Error same gender
	Error one is not adult
	Error one is already married
	Success

Agenda

- Introduction
- **MBT as a black box**
 - Introduction of the System Under Test
 - Test Objective Charter
 - Behavioral Model
 - Test Plan Publication
- Open the box
- The value of MBT
- Questions

Test Plan
(HTML)



Test Plan Publication (HTML)

Requirement	Aims	Tests
MARRIAGE <i>The conditions which allow a marriage are:</i> <ul style="list-style-type: none">- age over 18- none of them are married- one male and one female <i>Once the marriage is accepted, the status of the 2 inhabitants is modified accordingly their fields "Married with ..." is filled</i> <i>If an error occurs, an error message indicates which condition is not fulfilled.</i>	Error one is already married	testSuite_Family Marry_Error_one_is_already_married (02-23-c8)
	Error one is not adult	
	Error same gender	testSuite_Family Marry_Error_same_gender (02-9a-ed)
	Error same person	testSuite_Family Marry_Error_same_person (02-2a-e4)
	Success	testSuite_Family Marry_Success (02-ae-f1)
DIVORCE <i>The conditions which allow a divorce are:</i> <ul style="list-style-type: none">- 2 person being married together <i>Once the divorce is accepted, the status of the 2 inhabitants is modified accordingly: their fields "Married with ..." is empty.</i>	Error married but not together	testSuite_Family Divorce_Error_married_but_not_together (02-24-87)
	Error one is not married	testSuite_Family Divorce_Error_one_is_not_married (02-91-7c)
	Success	testSuite_Family Divorce_Success (02-57-2b)

Test Plan Publication (HTML)

Firefox ▾

Test: Marry_Error_same_person (02-2a-e4)

Steps	Actions	Requirements, aims
Step 1 (sut)	<u>Go to church</u> Click on the button "Go to church"	
Step 2 (sut)	<u>Marry</u> Click on "Get Married" Select the first person identified with EDDIE Select the second person identified with EDDIE Click on the bouton "OK"	REQ MARRIAGE AIM Error same person
2.1	<i>Check that the message ERROR_MARRIAGE_SAME_PERSON is displayed</i>	



Test Plan Publication (HTML)

Test Plan
(HTML)

Firefox ▾

Test: Marry_Error_one_is_already_married (02-23-c8)

Steps	Actions	Requirements, aims
Step 1 (sut)	<u>Go to church</u> Click on the button "Go to church"	
Step 2 (sut)	<u>Marry</u> Click on "Get Married" Select the first person identified with DOOLEY Select the second person identified with CHRISTELLE Click on the bouton "OK"	REQ MARRIAGE AIM Error one is already married
2.1	Check that the message <code>ERROR_MARRIAGE_NOT_SINGLE</code> is displayed	

Test Plan Publication (HTML)

Test Plan
(HTML)

Firefox ▾		
Test: Marry_Error_same_person (02-2a-e4)		
Steps	Actions	Requirements, aims
Step 1 (sut)	<u>Go to church</u> Click on the button "Go to church"	
Step 2 (sut)	<u>Marry</u> Click on "Get Married" Select the first person identified with EDDIE Select the second person identified with EDDIE Click on the bouton "OK"	REQ MARRIAGE AIM Error same person

Point Control Obs

Marry ✕

```
Click on "Get Married"
Select the first person identified with $IN_id1$
Select the second person identified with $IN_id2$
Click on the bouton "OK"
```

Precondition | Postcondition | Table Postcondition | Documentation

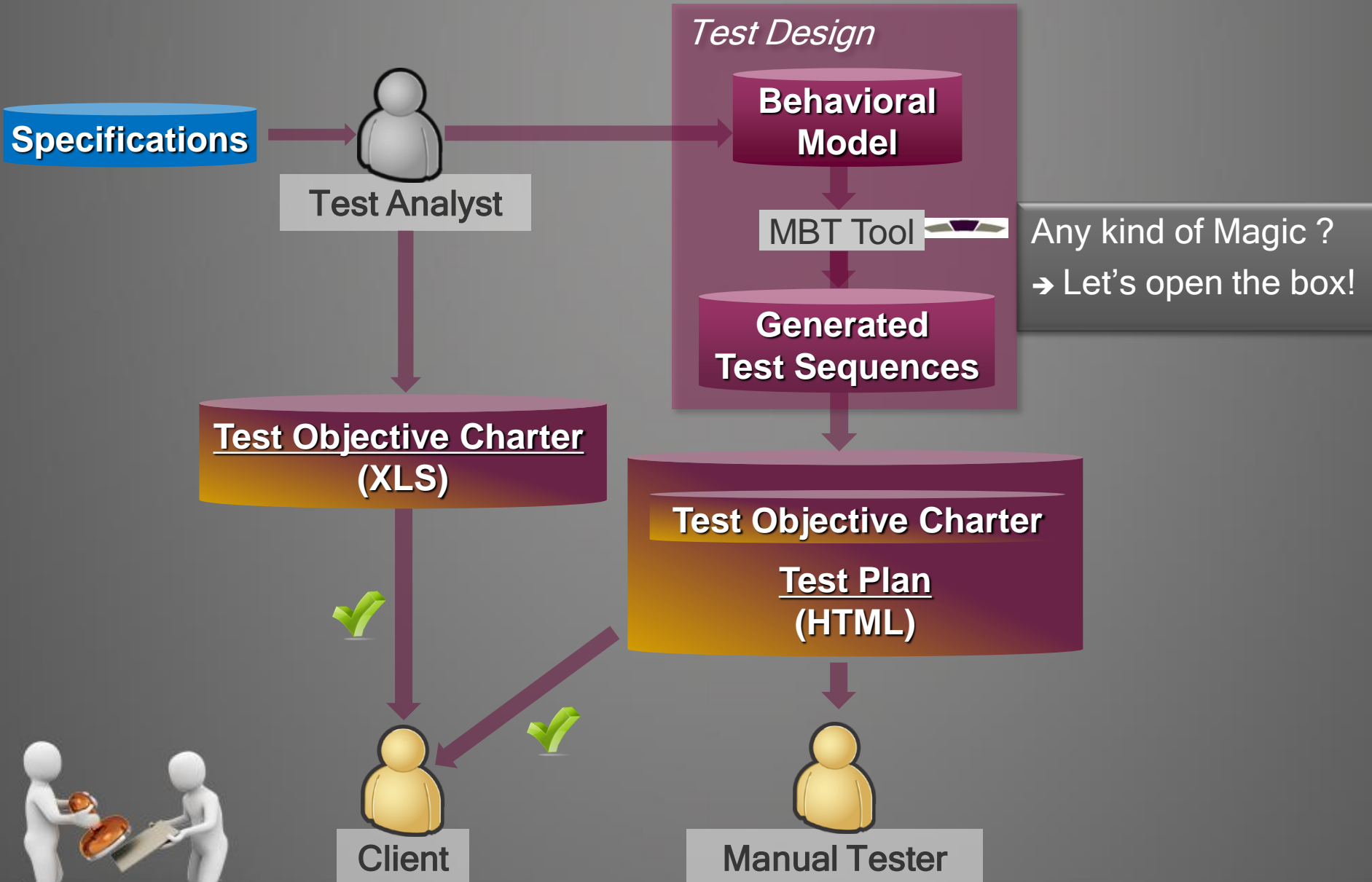
MARRIAGE_SAME_PERSON is displayed

Test: ..._is_already_married (02-23-c8)		
Requirements, aims		
Step 2 (sut)	<u>Marry</u> Click on "Get Married" Select the first person identified with DOOLEY Select the second person identified with CHRISTELLE Click on the bouton "OK"	REQ MARRIAGE AIM Error one is already married

2.1 Check that the message ERROR_MARRIAGE_NOT_SINGLE is displayed



MBT Process Overview

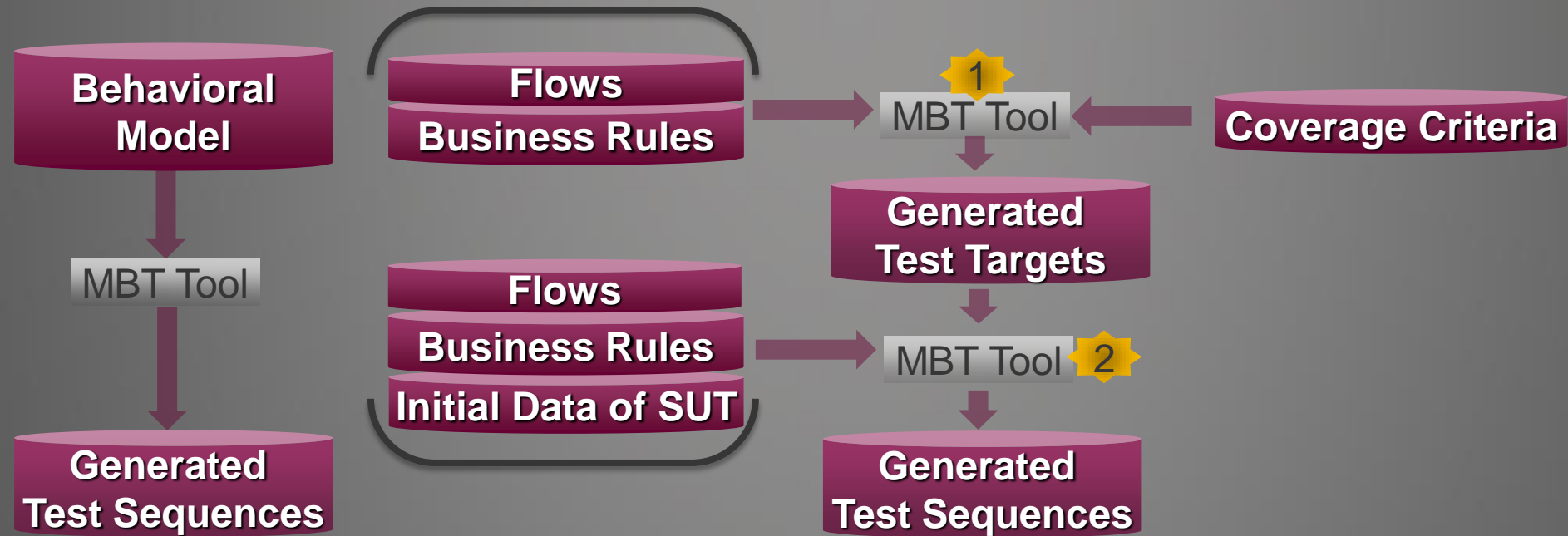


Agenda

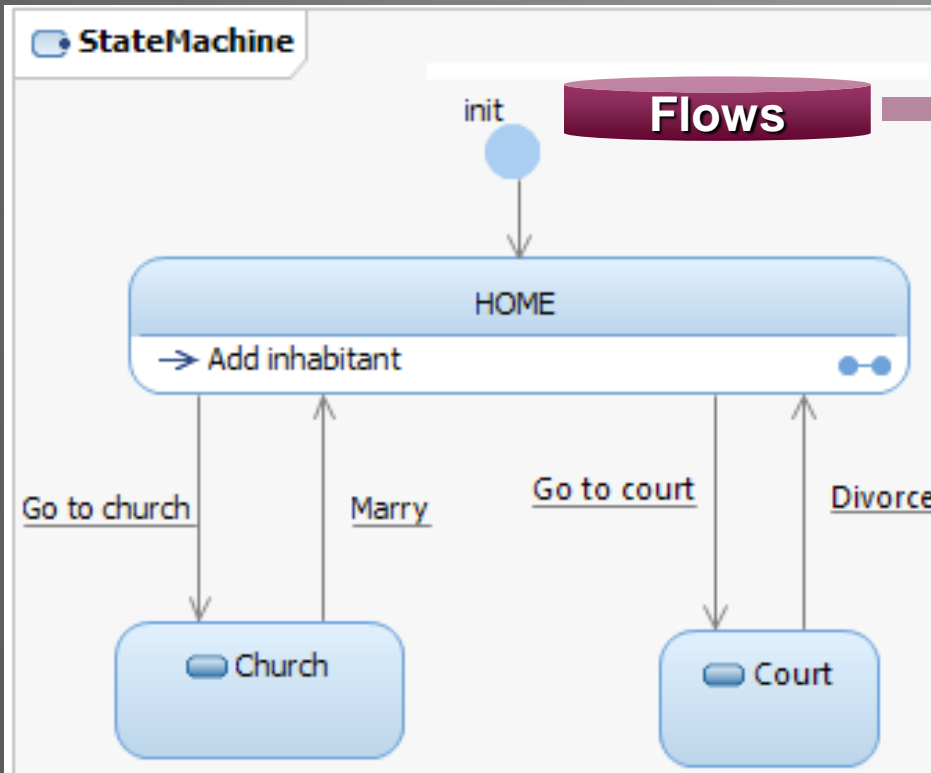
- Introduction
- MBT as a black box
- Open the box
- The value of MBT
- Questions



Test Generation Details



1 Test Target Generation from Model Flows



- Coverage Criteria
- Cover all transitions
 - Cover all states
 - ...



- Test Targets
- Cover Transition « Marry »
 - Cover Transition « Divorce »
 - ...



1

Test Target Generation from Business Rules

Coverage Criteria

- Cover all branches of the Rules
- Cover all conditions in a decision

Business Rules

```

---@REQ: MARRIAGE
if person1 = person2 then
  ---@AIM: Error same person
  self.message = MESSAGES::ERROR_MARRIAGE_SAME_PERSON

else if person1.gender = person2.gender then
  ---@AIM: Error same gender
  self.message = MESSAGES::ERROR_MARRIAGE_SAME_GENDER

else if (person1.age < 18) or (person2.age < 18) then
  ---@AIM: Error one is not adult
  self.message = MESSAGES::ERROR_MARRIAGE_NOT_AN_ADULT

else if person1.is_married() or person2.is_married() then
  ---@AIM: Error one is already married
  self.message = MESSAGES::ERROR_MARRIAGE_NOT_SINGLE

else
  ---@AIM: Success
  self.message = MESSAGES::NO_MESSAGE
  and
  if person1.gender = GENDER::MALE then
    ---@AIM: person1 is a man
    person1.wife = person2
  else
    ---@AIM: person1 is a woman
    person1.husband = person2
  endif
endif
  
```

1
MBT Tool

Coverage Criteria

Generated
Test Targets

Test Targets for « Marry »

- Cover Behavior « Error same person »
- Cover Behavior « Error same gender »
- Cover Behavior « Error one is not adult »
 - Only person1 is adult
 - Only person2 is adult
 - The 2 persons are not adult
- ...

2 Test Sequence Generation from Model Flows

Test Target:
- Cover the Transition « Marry »

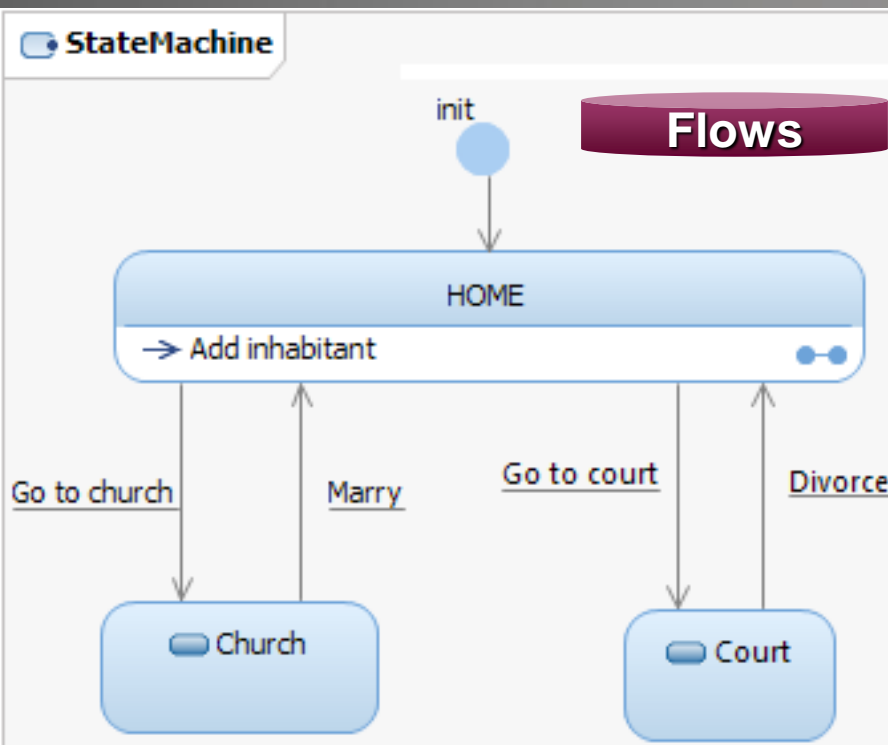
Generated
Test Targets

MBT Tool 2

Generated
Test Sequences

Test Sequence to cover that Test Target

- 1) Go to church
- 2) Marry

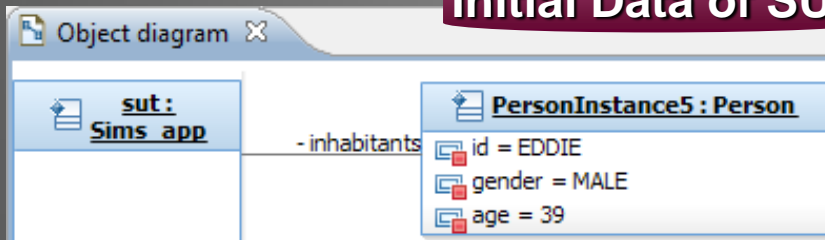


Test Sequence Generation From Business Rules

```
*Marry ✕
---@REQ: MARRIAGE
if person1 = person2 then
  ---@AIM: Error same person
  self.message = MESSAGES::ERROR_MARRIAGE_SAME_PERSON
```

Business Rules

Initial Data of SUT



Test Target:

- Transition « Marry »
- Behavior « Error same person »

Generated Test Targets

MBT Tool 2

Generated Test Sequences

Test Sequence to cover that Test Target

- 1) Go to church
- 2) Marry (
 - person1 = EDDIE,
 - person2 = EDDIE
)

Expected Result:

Message ERROR_MARRIAGE_SAME_PERSON



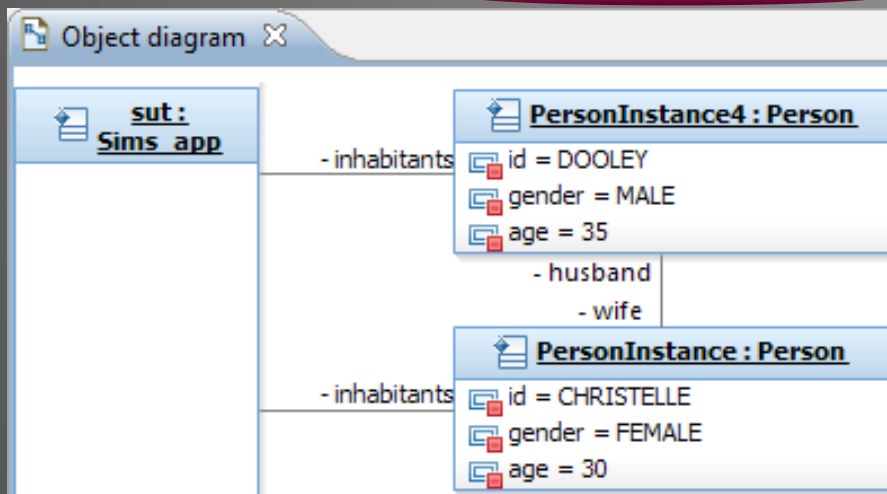
Test Sequence Generation From Business Rules

```

Marry x
else if (person1.is_married() or person2.is_married())=
  ---@AIM: Error one is already married
  self.message = MESSAGES::ERROR_MARRIAGE_NOT_SINGLE
  
```

Business Rules

Initial Data of SUT



Test Target:

- Transition « Marry »
- Behavior « Error one is already married »

Generated Test Targets

MBT Tool 2

Generated Test Sequences

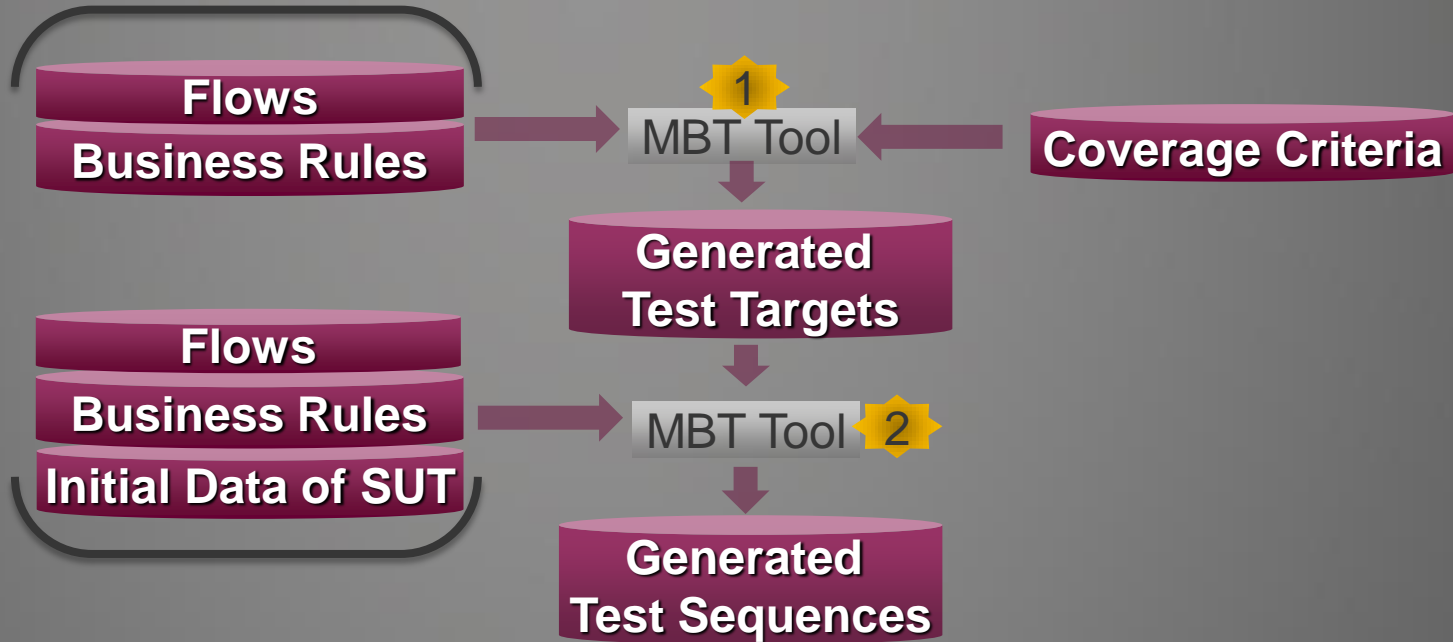
Test Sequence to cover that Test Target

- 1) Go to church
- 2) Marry (
 - person1 = DOOLEY,
 - person2 = CHRISTELLE

Expected Result:

Message ERROR_MARRIAGE_NOT_SINGLE

MBT Principles



Complexity ?

- MBT is dedicated to manage complexity and to reduce it
- Main key points are: Factorization and Reusability
- Let's see the impact of functional changes in the Model



Agenda

- Introduction
- MBT as a black box
- Open the box
- The value of MBT
- Questions



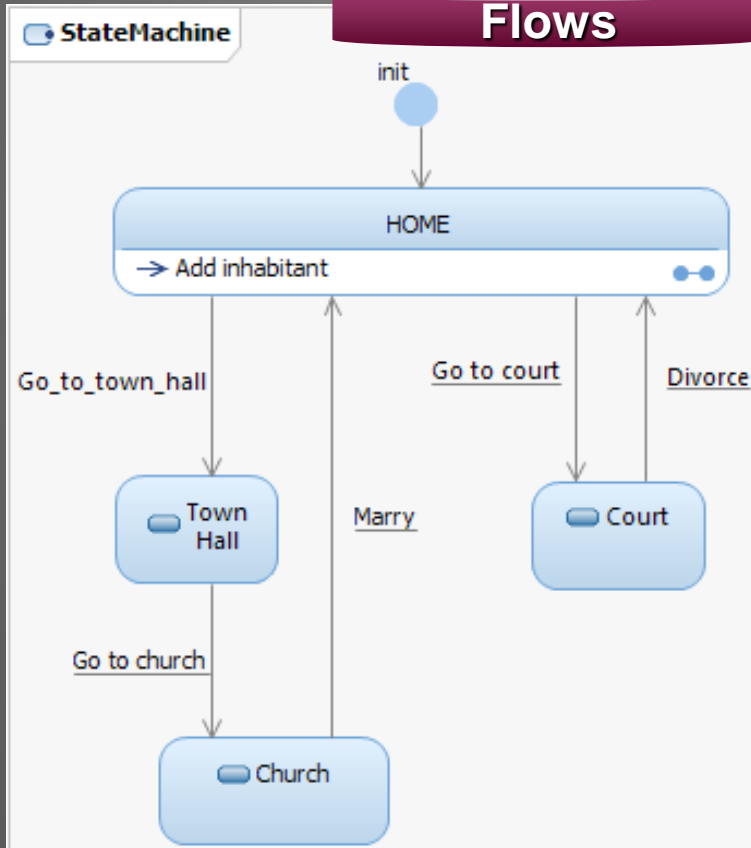
Change in Model Flows

Initial Data of SUT

Business Rules

Flows

MBT Tool



Test Sequence

- 1) Go to town hall
- 2) Go to church
- 3) Marry (
 person1 = EDDIE,
 person2 = EDDIE
)

Expected Result:

Message ERROR_MARIAGE_SAME_PERSON

Test Sequence

- 1) Go to town hall
- 2) Go to church
- 3) Marry (
 person1 = DOOLEY,
 person2 = CHRISTELLE
)

Expected Result:

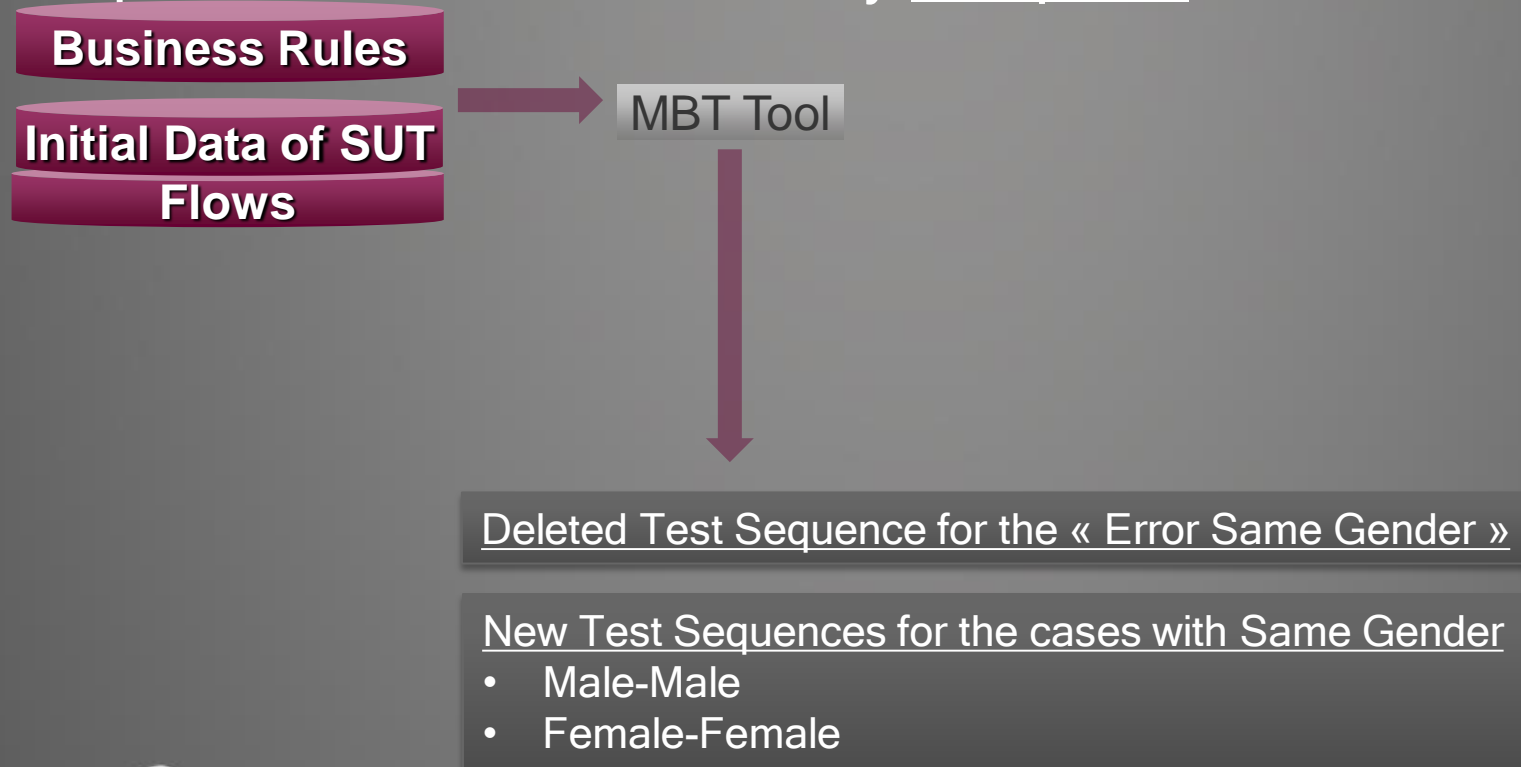
Message ERROR_MARIAGE_NOT_SINGLE



Change in Business Rules

« Marriage between same gender now Allowed »

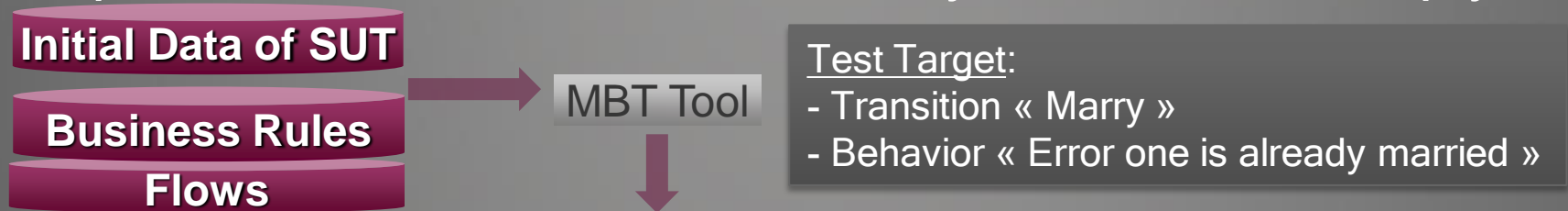
→ Impact in Business Rules only at 1 place



Change in Initial Data of SUT

Do not want to define any pre-requisites for Initial Data

→ Impact in « Initial Data of SUT » only which is now empty



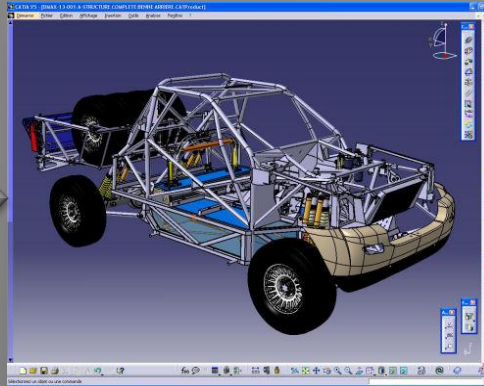
Updated Test Sequence to cover that Test Target

- 1) Add_Inhabitant (id = ID1, gender = MALE, age = 20)
- 2) Add_Inhabitant (id = ID2, gender = FEMALE, age = 21)
- 3) Go to church
- 4) Marry (person1 = ID1, person2 = ID2)
Expected Result: Message SUCCESS
- 5) Go to church
- 6) Marry (person1 = ID1, person2 = ID2)
Expected Result: Message ERROR_MARRIAGE_NOT_SINGLE
- 7) Delete_Inhabitant (ID1)
- 8) Delete_Inhabitant (ID2)



As Take Away

- The metaphore for MBT



- Factorization
- Reusability
- Capitalization
- Productivity
- Change Management

- Other benefits of MBT

- Regarding specifications, Model is closer than Tests
- Early raise issues in specifications
- Booster for test automation of the tests



Agenda

- Introduction
- MBT as a black box
- Open the box
- The value of MBT
- Questions



Thank your for attention !
Any question ?



Eddie.Jaffuel@eConsult.fr