Muhis511, Muhammad Ismail, Lab: 4, Model-based Testing

Part 1:

Yes it detects now by adding "**random(never or time_duration(120)"** this line I use to detect results in graph walker.

I added assert code that fail the assertion if it visit an edge twice.

e_AnotherAction edge I added a line of code **assertTrue(visit);** after visiting first time a set false the visit variable and it fails on next visit.

I use command mvn graphwalker:test to check it in console.

What is the relationship between the model coverage, as in edge coverage, vertex coverage or path coverage when GraphWalker executes the tests that implement the model steps, and code coverage of the application code?

Model coverage is related to branch or code that represent it. Edge & vertex coverage is related to statements and it use to represent statement of a system.

Does 100% edge coverage imply 100% code coverage? Does it imply 100% branch coverage?

Nope, 100% edge coverage doesn't mean it test or cover all of the code. In some detailed test cases in which it visit all statement (edges) can possible to cover all code coverage. That is difficult to implement.

Edge coverage if it covering all the branches of system then it is equal to the branch coverage. So we can say 100 edge coverage means 100% branch coverage.

Part 2:

Changes:

I introduce two changes

```
@Column(name = "city")
@NotEmpty
@Size(min = 3, max = 10)
private String city;

@Column(name = "telephone")
@NotEmpty
@Pattern(regexp="\\+[0-9]+$")
@Digits(fraction = 0, integer = 10)
private String telephone;
```

->Commit (0922337 City name max, Number start with +, changes by muhis511)

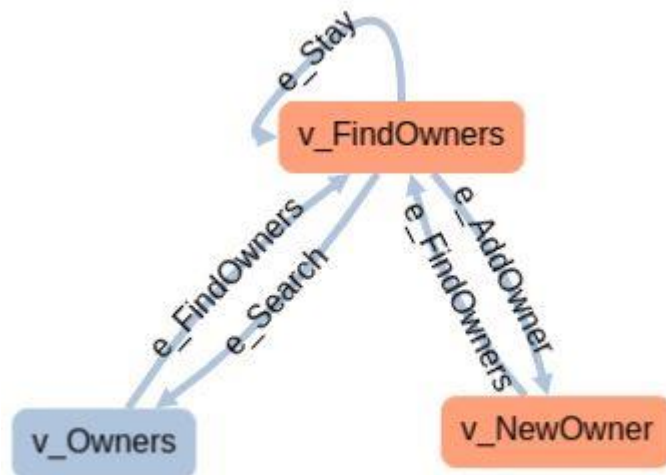1: Add limit constraints to city name in owner form, input like min=3 and max=10

2: Add + is mandatory in telephone number of owner form, by adding regexp as shown in above image.

I changed these and push to git by using (muhis511) id. And share repository with my partner adith054.

Fixed:

1:

He did changes in Search owner constraint by making it required. So the graph walkers was failing when searching owner with empty search. I fixed it by creating and edge (e_Stay) and it keeping it stay, if search is empty. You can check it in File FindOwnerTest.java.

2:

Second change that my partner did was in owner form firstName constraint. He make it like, it doesn't accept number and accept only Alphabets. I fixed it by making an edge (e_IncorrectFirstName) that if wrong go to the vertex (v_IncorrectData) that through it back to the same page. And if data is correct it moves to next state.