Applying Mutation Testing (MT) to Event Processing Language (EPL)

L. Gutiérrez-Madroñal

July, 2015

Index

Content of the presentation

1 MT and real-time relationship

2 Real-time evolution

3 Event Processing Language

4 EPL MT architecture

5 Results

6 Conclusions

Index

Content of the presentation

1. MT and real-time relationship
2. Real-time evolution
3. Event Processing Language
4. EPL MT architecture
5. Results
6. Conclusions

MT and real-time relationship

- MT has been applied to many traditional programming languages that now are applied to real-time systems: Java, C, Ada...
- Some studies are focus on real-time systems: models for generating test cases for testing of timeliness...

Index

Content of the presentation

1 MT and real-time relationship

2 Real-time evolution

3 Event Processing Language

4 EPL MT architecture

5 Results

6 Conclusions

Real-time evolution

- “Complex Event Processing” (CEP): analyzes series of events for deriving conclusions
- MT study of “Event Processing Language” 4.9.0. (EPL): a domain-specific language for processing events

Index

Content of the presentation

1. MT and real-time relationship
2. Real-time evolution
3. Event Processing Language
4. EPL MT architecture
5. Results
6. Conclusions

Event Processing Language

Definition

Is a SQL-like language with SELECT, FROM, WHERE, GROUP BY, HAVING and ORDER BY clauses, but differs from SQL in its use of views rather than tables. EPL operates on continuous stream of events.

Listing 1: EPL example

```
select A as temp1, B as temp2 from pattern
[every temp1.temperature > 400 ->
temp2.temperature > 400]
```

Index

Content of the presentation

1 MT and real-time relationship

2 Real-time evolution

3 Event Processing Language

4 **EPL MT architecture**

5 Results

6 Conclusions

EPL MT architecture

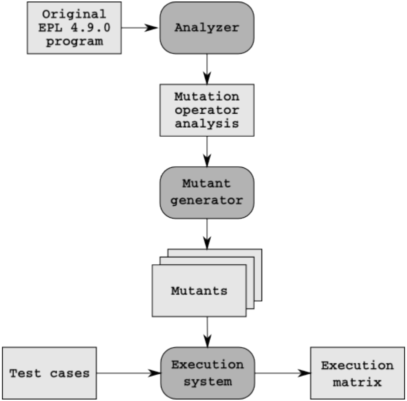

Figure : Basic architecture

EPL MT architecture

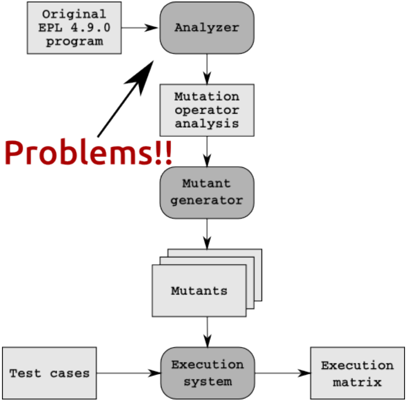

Figure : Analysis problem

EPL MT architecture

Analysis problem

Some of the values of an EPL query are obtained during the execution time.

EPL capturer

The architecture includes a component which captures them wholly *(Execution time)*.

EPL MT architecture

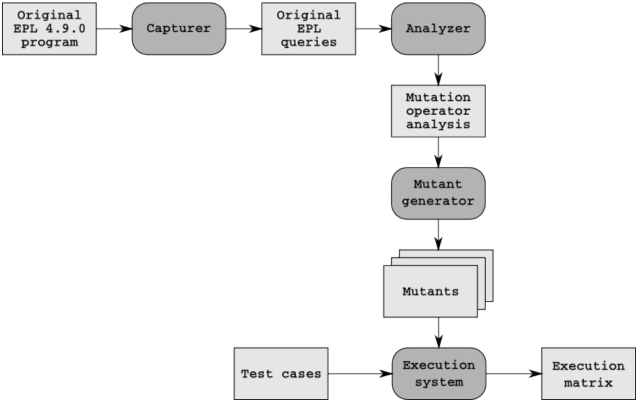

Figure : Architecture with EPL capturer

EPL MT architecture

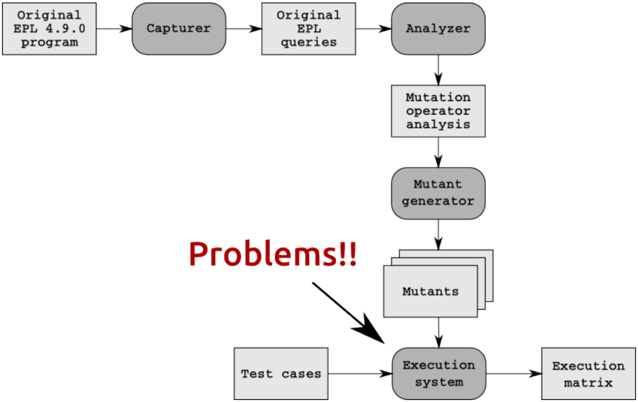

Figure : Execution problem

EPL MT architecture

Execution problem

Due to the data nature which the architecture is dealing with - *events in real-time*. We must ensure that **all the programs recieve the same events**.

Parallel execution

- The execution system includes a mechanism that can **synchronise the execution threads**.
- This mechanism builds a *barrier* where all threads must wait, until all threads reach it, before any of the threads can continue.

EPL MT architecture

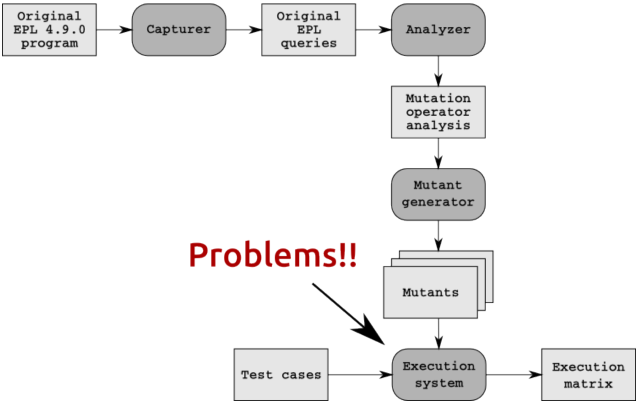

Figure : Execution problem - **Not solved!!**

EPL MT architecture

Execution problem

It has been verified that the outputs are subject to the machine in which the programs are executed as well as the system time.

Test case generator

- It is needed to obtain the **event values** to ensure that the programs recieve the same events.
- **Custom test cases** are obtained.

EPL MT architecture

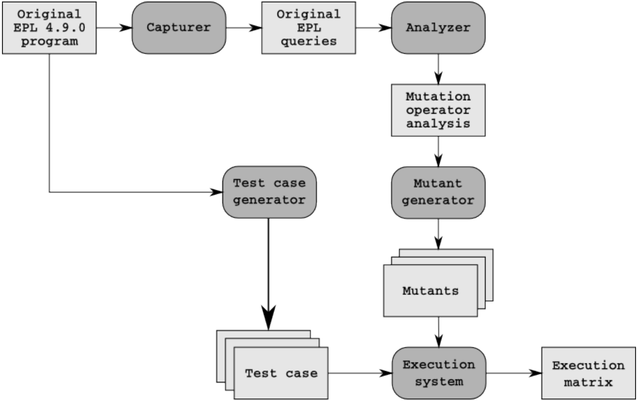

Figure : Test case generator

Index

Content of the presentation

1 MT and real-time relationship

2 Real-time evolution

3 Event Processing Language

4 EPL MT architecture

5 Results

6 Conclusions

Results

So far, programs under test:

- Smaller executions show more live mutants
- Longer executions show more wrong mutants

Index

Content of the presentation

1 MT and real-time relationship

2 Real-time evolution

3 Event Processing Language

4 EPL MT architecture

5 Results

6 Conclusions

Conclusions

- Several requirement have to be taking into account to develop an EPL architecture
- The interesting executions are the smaller ones

Thanks!

Questions?

L. Gutiérrez-Madroñal
lorena.gutierrez@uca.es

Presentation made by LaTeX template from www.godtic.com