Generating Complex Metamorphic Relations for Cyber-Physical Systems with Genetic Programming

Jon Ayerdi∗, Valerio Terragni †, Aitor Arrieta∗,
Paolo Tonella ‡ and Maite Arratibel §

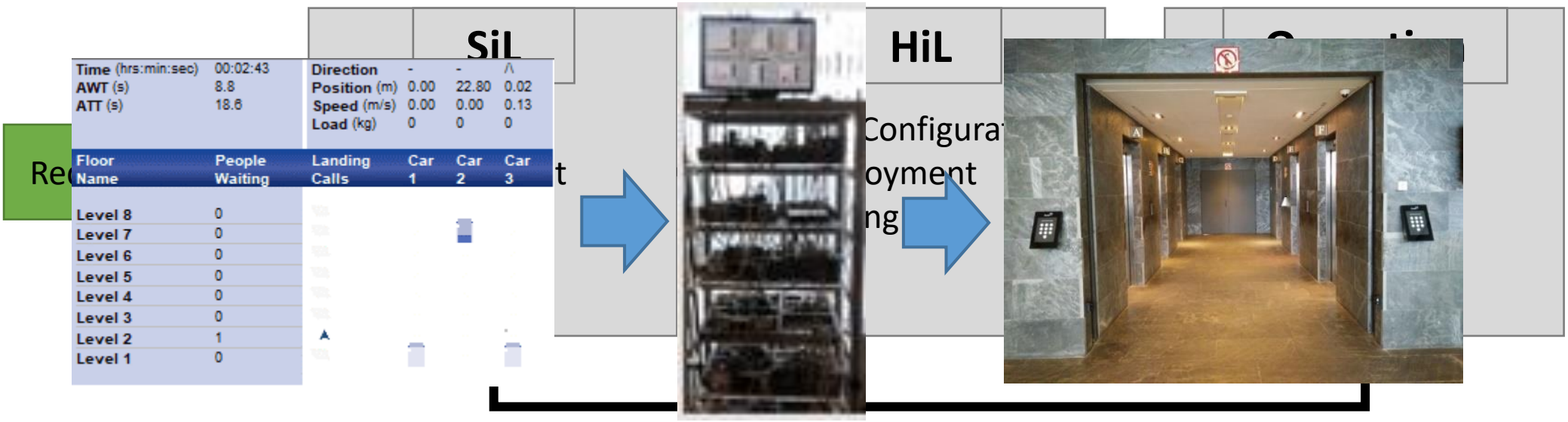
University of Mondragon∗, University of Auckland †,
Università della Svizzera italiana (USI) ‡, Orona §

jayerdi@mondragon.edu

Orona

- Leading elevator company in Europe
- Design, manufacturing, installation and maintenance of:
  - Elevators
  - Escalators
  - Moving ramps
  - …
- Multi-elevator installations

Software maintenance process



- New releases take an average of around **a year to complete this process**
- Many of these steps are **not fully automated**
- Notably, major reliance on **human test oracles**

Metamorphic Testing

- Based on the relations among the inputs and outputs of **two or more test executions**, the so called metamorphic relations (MRs)

- MRs for the domain of elevation already defined in our previous work [1]

$$E_f = E_s \cup E' \Rightarrow AWT_f < AWT_s$$

[1] Ayerdi et al. QoS-aware Metamorphic Testing: An Elevation Case Study. ISSRE 2020.

Automatic generation of MRs

- MRs have been proven effective
- Manual definition of MRs can be **costly and error-prone**
  - Requires in-depth knowledge of the domain and the system

- Build on top of GAssert, a technique for **automatically generating** and improving **program assertions** [1]

GAssertMRs, Genetic ASSERTion improvement for MRs [2]

[1] Terragni et al. Improving Assertion Oracles with Evolutionary Computation. ESEC/FSE 2020.

[2] Ayerdi et al. Generating Metamorphic Relations for Cyber-Physical Systems with Genetic Programming: an Industrial Case Study. ESEC/FSE 2021.

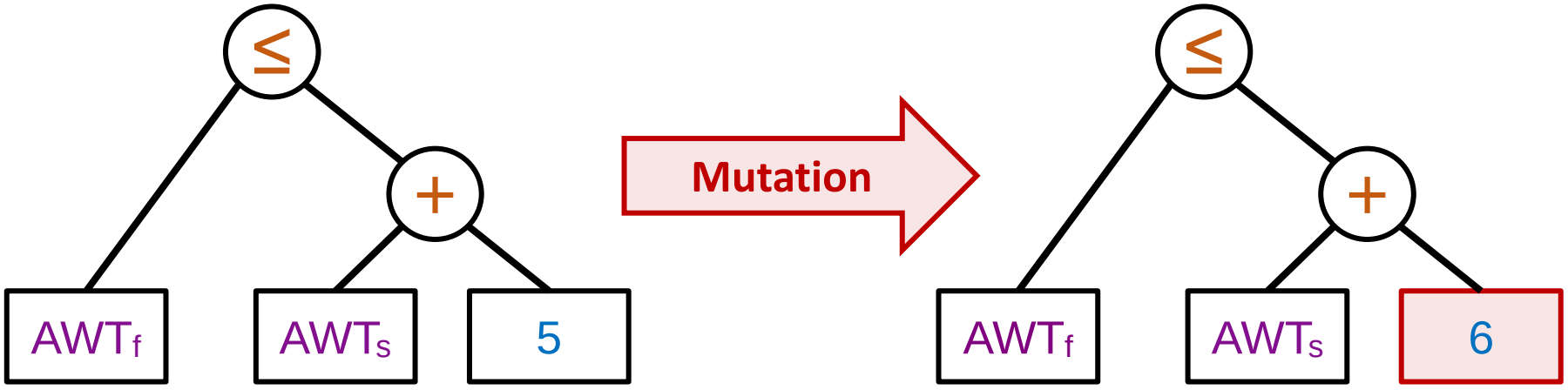
Evolutionary Algorithm

- Fitness for MRs: Minimize **FPs**, **FNs** and **complexity**
  - Usually evaluated with **mutation testing**
  - Real failures could be used if available

- Template for performance/QoS related MRs:
  - $\langle METRIC_f \rangle$ <operator> <expression>
  - $AWT_f \leq AWT_s + 5$
- Only the <expression> part is generated by GAssertMRs
  - Reduced **search space**
  - MRs are **easier to understand** by humans

Genetic Programming

- Individuals **represented as expression trees**
- Expressions can contain operators, variables and constants
  - Operators: Arithmetic (+, /, …), relational (<, =, …) or logical (AND, OR, …)
  - Variables: System inputs or outputs
  - Constants: Numeric (-100, 100) or Boolean (true/false)



$AWT_f \leq AWT_s + 5$

$AWT_f \leq AWT_s + 6$

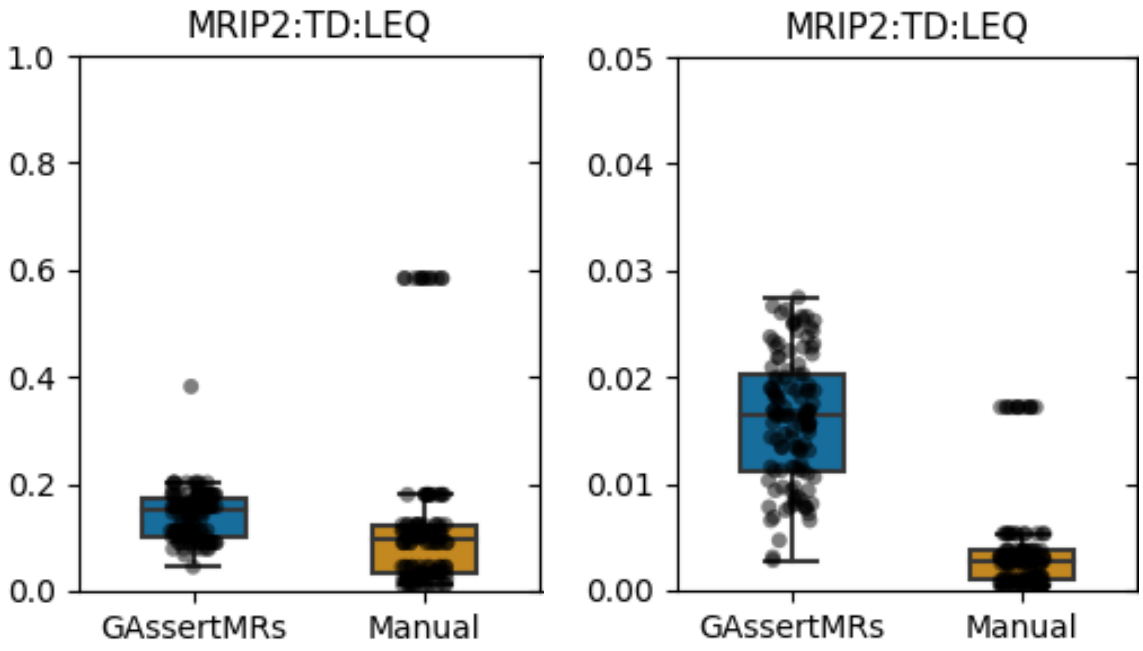
Evaluation – Elevator Installations (Orona)

- QoS metrics
  - Average Waiting Time (AWT)
  - Total Distance (TD)
  - Total Movements (TM)

- Evaluation
  - Configurations: Try every QoS metric and operator (> or <) combination
  - **Learning** process took **15 mins** for each configuration

Evaluation – Experimental Results

- GAssertMRs results in **almost zero FPs**
- **Outperforms manual MRs** in several configurations
- **Outperforms regular assertions** except for one configuration



Mutation Score

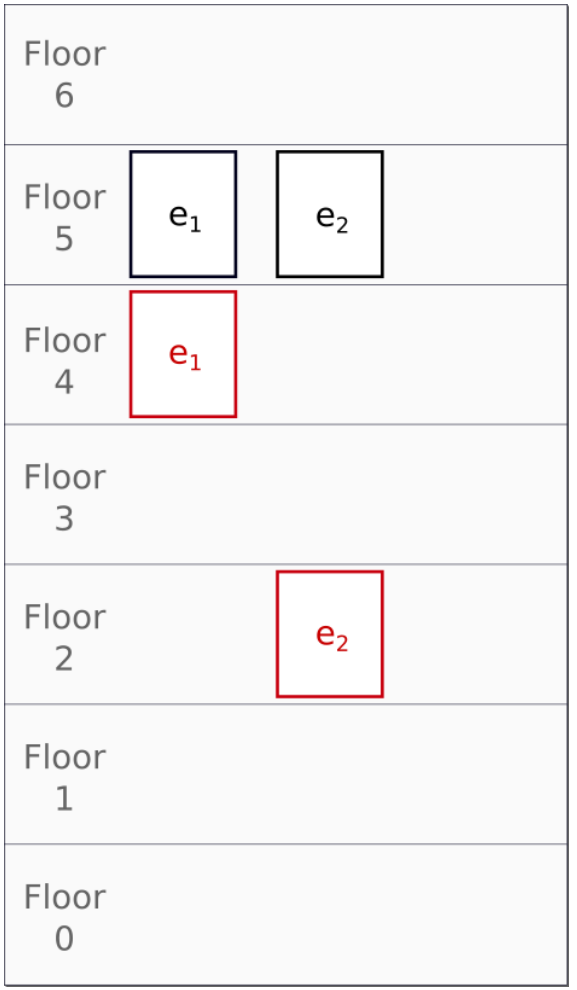
Failure Detecion Ratio

Future Extension – Complex Types

- Use **functional programming techniques** to enable the manipulation of collections
  - Function literals
  - Higher order functions

positionsDistance(E_s, E_f)

sum(map([(e_s, e_f)]{abs(e_s − e_f)}, zip(E_s, E_f)))



[(5,4), (5,2)] → [1,3] → 4

Mondragon Unibertsitatea

Faculty of Engineering

Thank you

Jon Ayerdi

jayerdi@mondragon.edu