

Alf-verifier : An Eclipse Plugin for Verifying Alf/UML Executable Models*

Elena Planas
eplanash@uoc.edu

David Sánchez
dsanchezmen@uoc.edu

Jordi Cabot
jordi.cabot@inria.fr

Cristina Gómez
cristina@essi.upc.edu



What is Alf-verifier?

Alf-verifier is an **Eclipse plugin** that implements a lightweight method for **verifying** if the **operations** attached to the classes of a UML class diagram are **Strongly Executable**.

An operation is **Strongly Executable (SE)** if it is always successfully executed, i.e. if every time we execute the operation, the effect of the actions included in the operation evolves the initial state of the system to a new system state that satisfies all integrity constraints of the structural model.

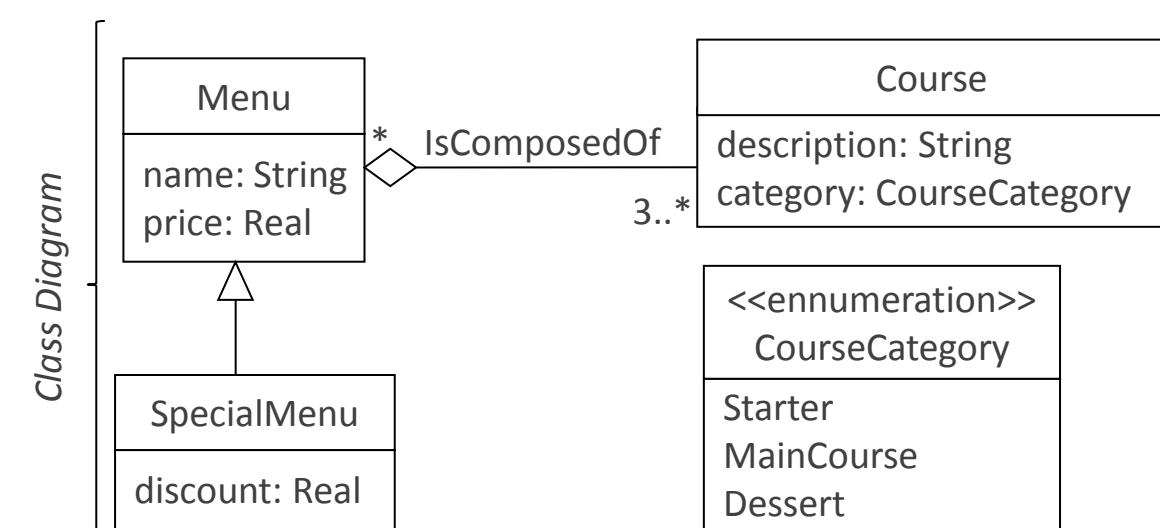
INPUT

1 Specify the UML executable model to be verified.

Alf-verifier takes as input an **executable model** composed by:

- (1) A **UML class diagram** and a set of **OCL integrity constraints**.
- (2) A set of operations specified by means of **Alf** ← an OMG-standard action language to specify detailed behavioural models.

Structural Model



Integrity Constraints

- (1) context Menu inv: Menu.allInstances()->isUnique(name)
- (2) context SpecialMenu inv: SpecialMenu.allInstances()->size()≤3
- (3) context SpecialMenu inv: self.discount≥10

Behavioural Model

```
activity classifyAsSpecialMenu (in _discount: Real) {  
  if ( _discount >= 10 ) {  
    classify self to SpecialMenu;  
    self.discount = _discount;  
  }  
}
```

The operation **classifyAsSpecialMenu** is **not strongly executable** since after its execution the maximum integrity constraint of class **SpecialMenu** may be violated

2 Select the operation to be verified and invoke the verifier.

Alf-verifier

OUTPUT

3 Alf-verifier returns either:

- ✓ A positive answer, if the input operation is **Strongly Executable**
- ✗ A negative answer + a corrective **FEEDBACK**, if the input operation is **NOT SE**

The feedback consists in a set of **actions** and **preconditions** that should be added to the operation in order to make it strongly executable.

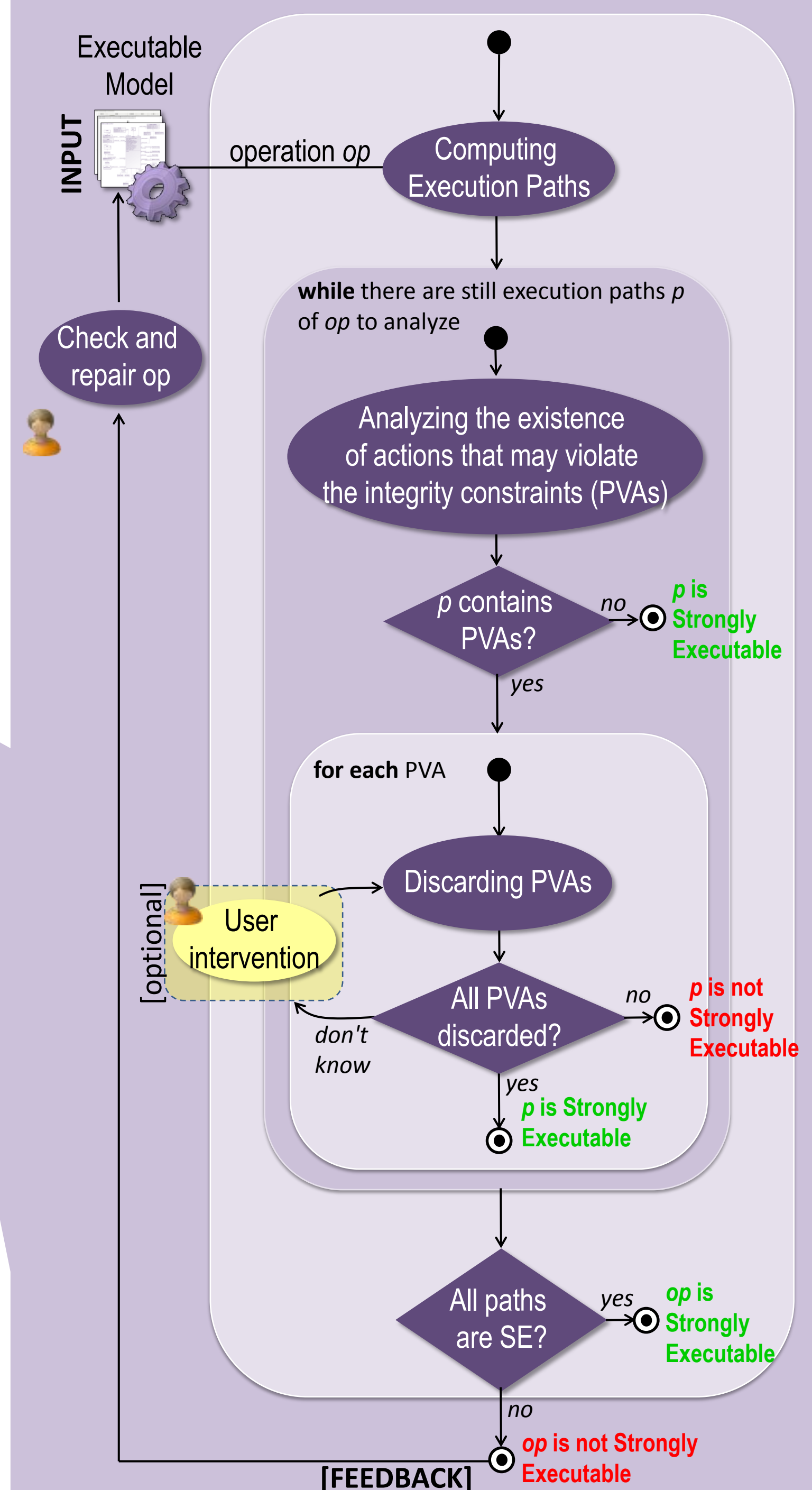
In order to avoid the violation of the constraint context SpecialMenu inv: SpecialMenu.allInstances()->size()≤3, a precondition ensuring there are less than three special menu should be added.

```
activity classifyAsSpecialMenu (in _discount:Real){  
  if ( _discount >= 10 and SpecialMenu.allInstances()->size<3 ){  
    classify self to SpecialMenu;  
    self.discount = _discount;  
  }  
}
```

Now, the operation **classifyAsSpecialMenu** is **strongly executable**

How it internally works?

It performs an **static analysis** (i.e. without executing the operation) of each execution path of the operation.



* Available at

<http://code.google.com/a/eclipseorg/p/alf-verifier/>