Preuve unitaire d'un logiciel avionique

« Preuve de modèle, preuve de programme »

-

CYCLE DE CONFÉRENCES TECHNIQUES SUR LES MÉTHODES FORMELLES DE DÉVELOPPEMENT

Stéphane Duprat (AtoS) stephane.duprat@atos.net

04.02.2014

Agenda

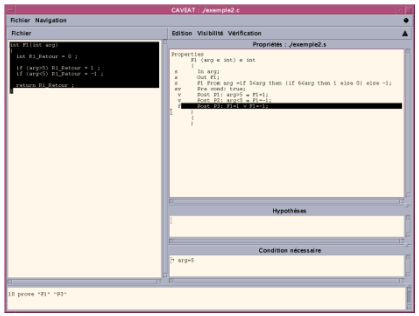
- Proof of program overview
- Unit Proof & Integrated Proof
- Unit Proof objectives
- Unit Proof solution
- Unit Proof results
- Conclusion

Introduction

Industrial feedback of Program verification of Airbus avionics software.

AIRBUS

Tooling

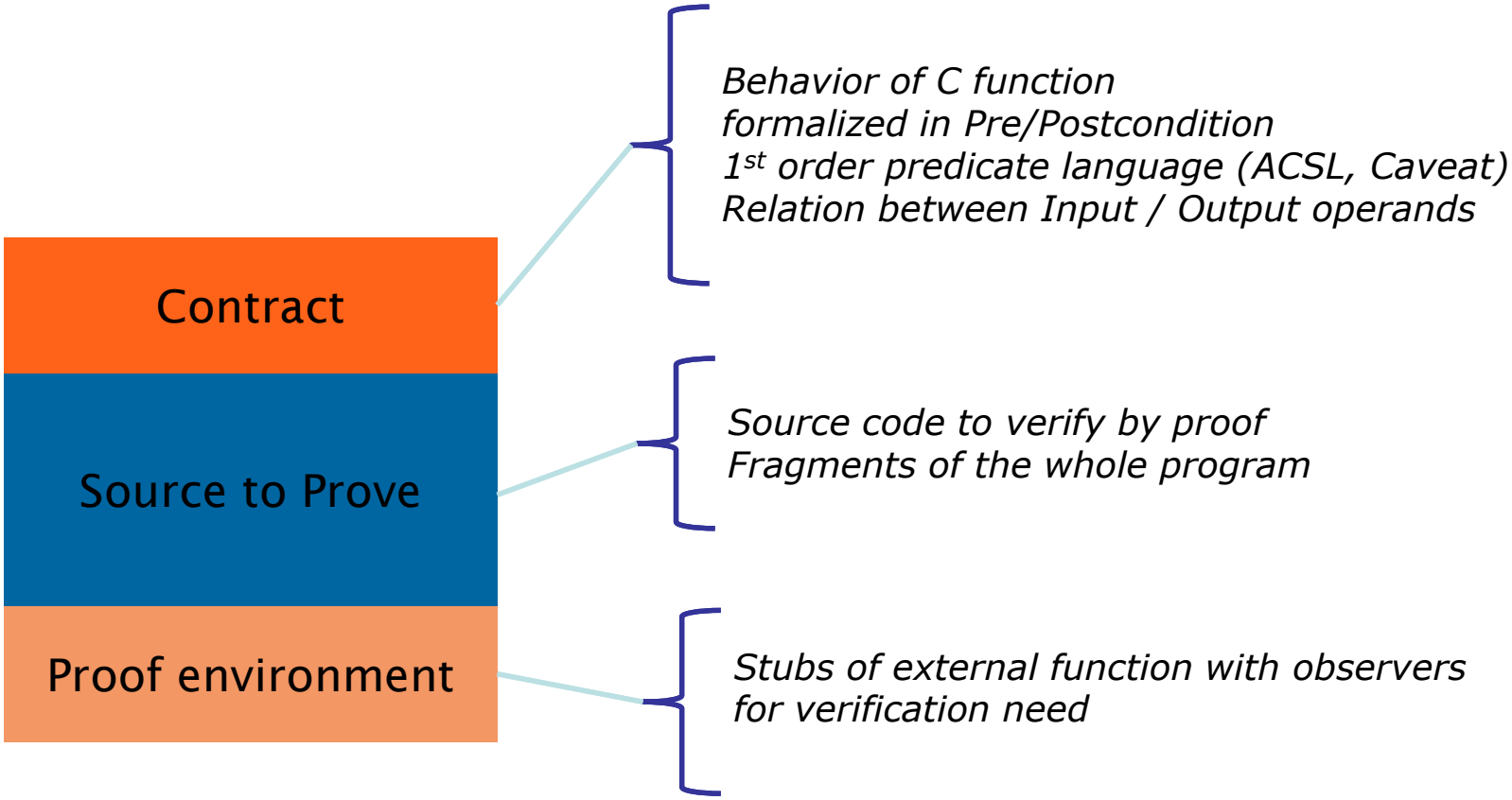


Caveat

frama C
Software Analyzers

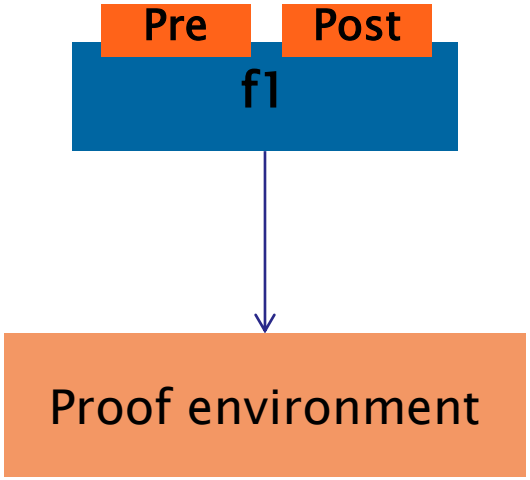
Frama-C/WP

Proof of program overview

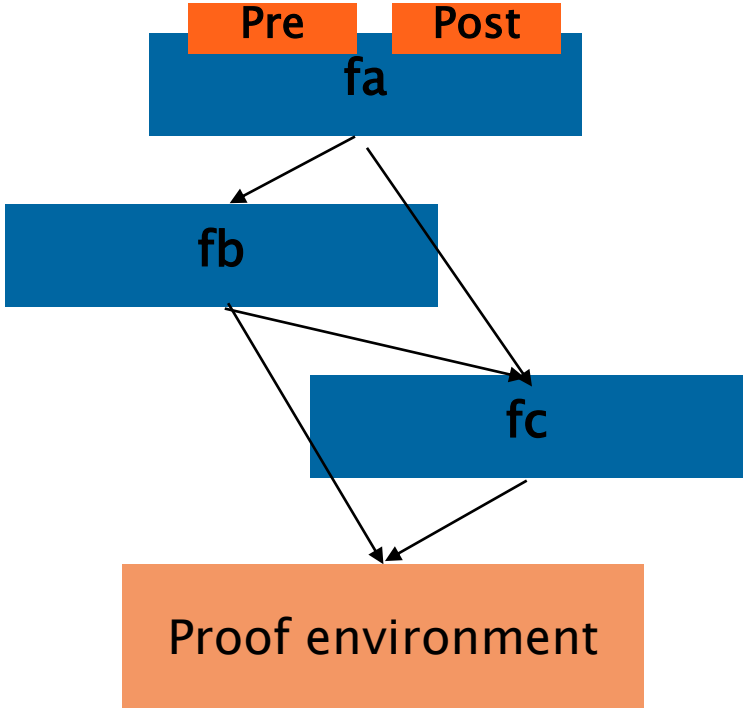


Unit Proof & Integrated Proof

- Unit Proof



- Integrated Proof



Integrated Proof / Example


```
int MyTab[10] ;
int S;

extern int SumElt(int s, int i);
extern int IsEven(int n);
```

```
Const f_even ∈ int -> Bool =
  lambda n ∈ int. (n modulo 2) = 0;

Const f_sum ∈ int -> Tab&int → int ;

Axiom AX1_sum :
  forall n ∈ int, t ∈ Tab&int.
     n<0  => f_sum(n, t) = 0 ;

Axiom AX2_sum :
 forall n ∈ int, t ∈ Tab&int.
  n>=0 => f_sum(n, t) = f_sum(n-1, t) + (if f_even(t.[.(n)]) then 1 else 0) ;
```

Post P1

```
void SumTab(void)
{
  int i;
  S=0;

  for (i=0; i<10; i++)
    S = SumElt(S, i);
}
```

Inv CPT1, SET1

```
FONCTION SumTab

Post P1 : S = f_sum(9,MyTab);

Inv 1 I1_CPT1 : i>=0 && i<10 ;
Inv 1 I1_SET1 : S = f_sum(i-1, MyTab);
```

Pre PR1, PR2

Post P1

```
int SumElt(int s, int i)
{
  int sum = s;

  if (IsEven(MyTab[i]))
    sum++;

  return sum ;
}
```

```
FONCTION SumElt

Pre  PR1 : s = f_sum(i-1, MyTab);
Pre  PR2 : i>= 0 ;
Post P1  : SumElt = f_sum(i, MyTab);
```

Post P1

```
int IsEven(int n)
{
  return !(n % 2);
}
```

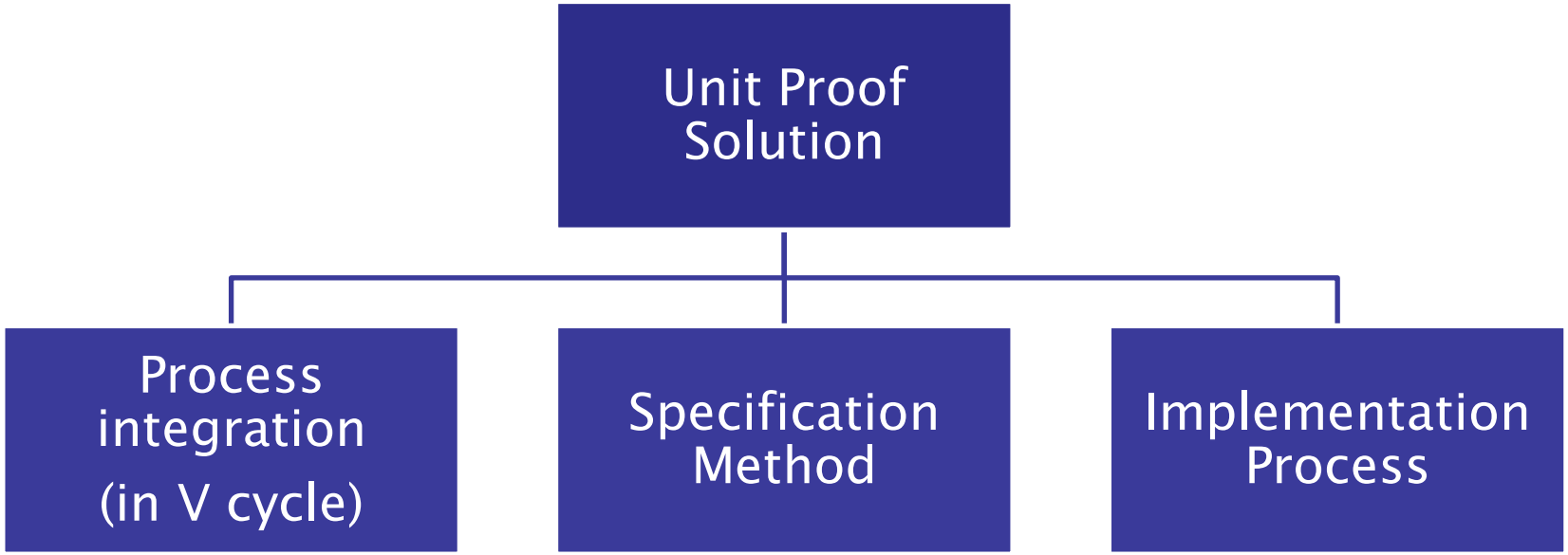
```
FONCTION IsEven

Post P1 : IsEven = if (f_even(n)) then 1 else 0;
```

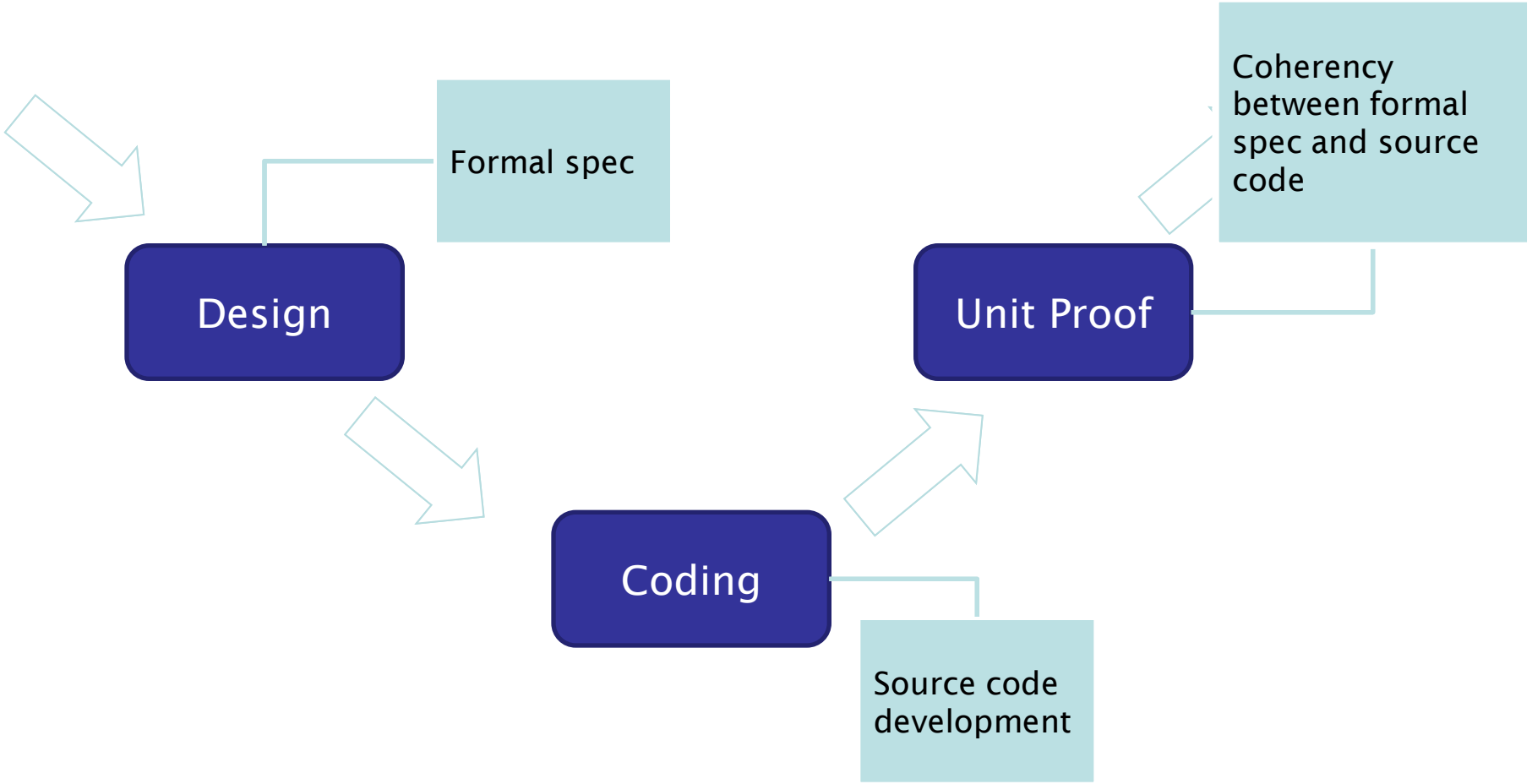
Unit Proof objectives

- Functional verification of C function (replacing Unit Testing)
- Achievable for standard developer, non-specialist of formal method
- Cost Effective
- Maximal automaticity
- Quality equivalent (compared with tests)

Unit Proof Solution



Unit Proof Solution
Process Integration



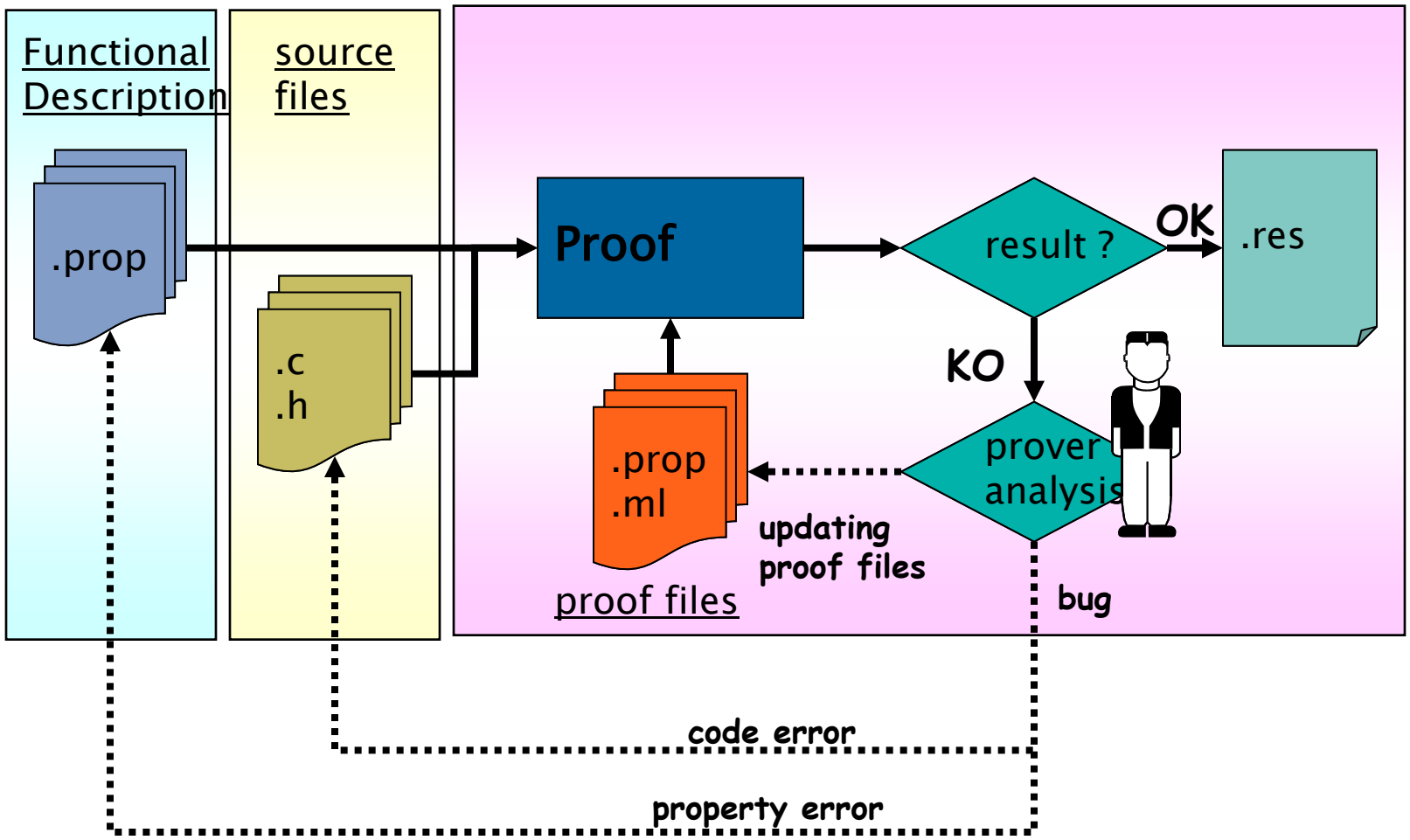
Unit Proof Solution
Specification Method

- Specification of
  - All behaviors
  - All operands
    - Function parameter
    - Global variables
    - In / Out operands of functions called
    - Volatile access
  - Sequence of function calls
  - Constant values

- Definition of a standard way of specification
  - Allowing check activities
  - Easy to read

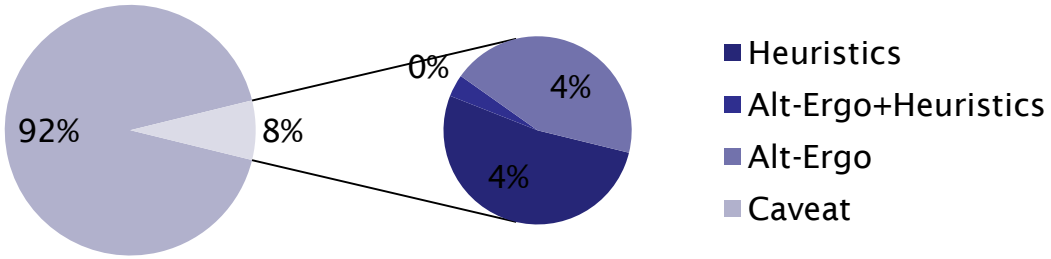
Unit Proof Solution

Implementation Process



Unit Proof Results

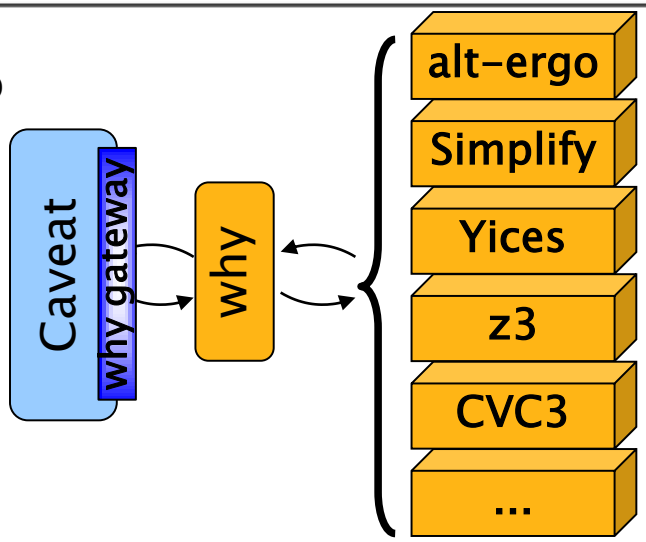
- Since 2006
- 3 Aircraft Programs (A380, A400M, A350)
- ~ 40 kloc of C source Code
- ~ 5000 Postconditions
- Proof campaign : ~1 day
- Automaticity



- Ease of maintenance
- No problem of skills in development team

Tooling Improvement

- Caveat Update with AltErgo



- Frama-C NUPW
  - Import from Caveat
  - Improvement for proof
  - Wider area of C programs
  - ACSL specification

Thank you

04.02.2014