Informatics Department
University of Minho

Ada Europe

2012

An Approach to Model Checking Ada Programs

Authors: João Martins, José Miguel Faria and Jorge Sousa Pinto

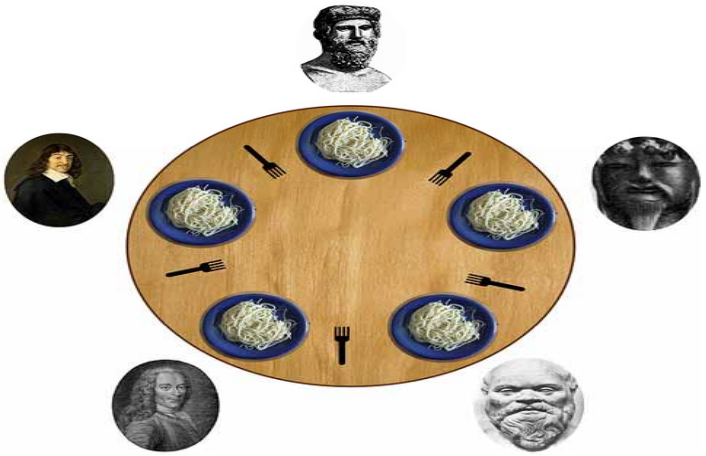
Stockholm, 13th June, 2012

Introduction and Motivation I

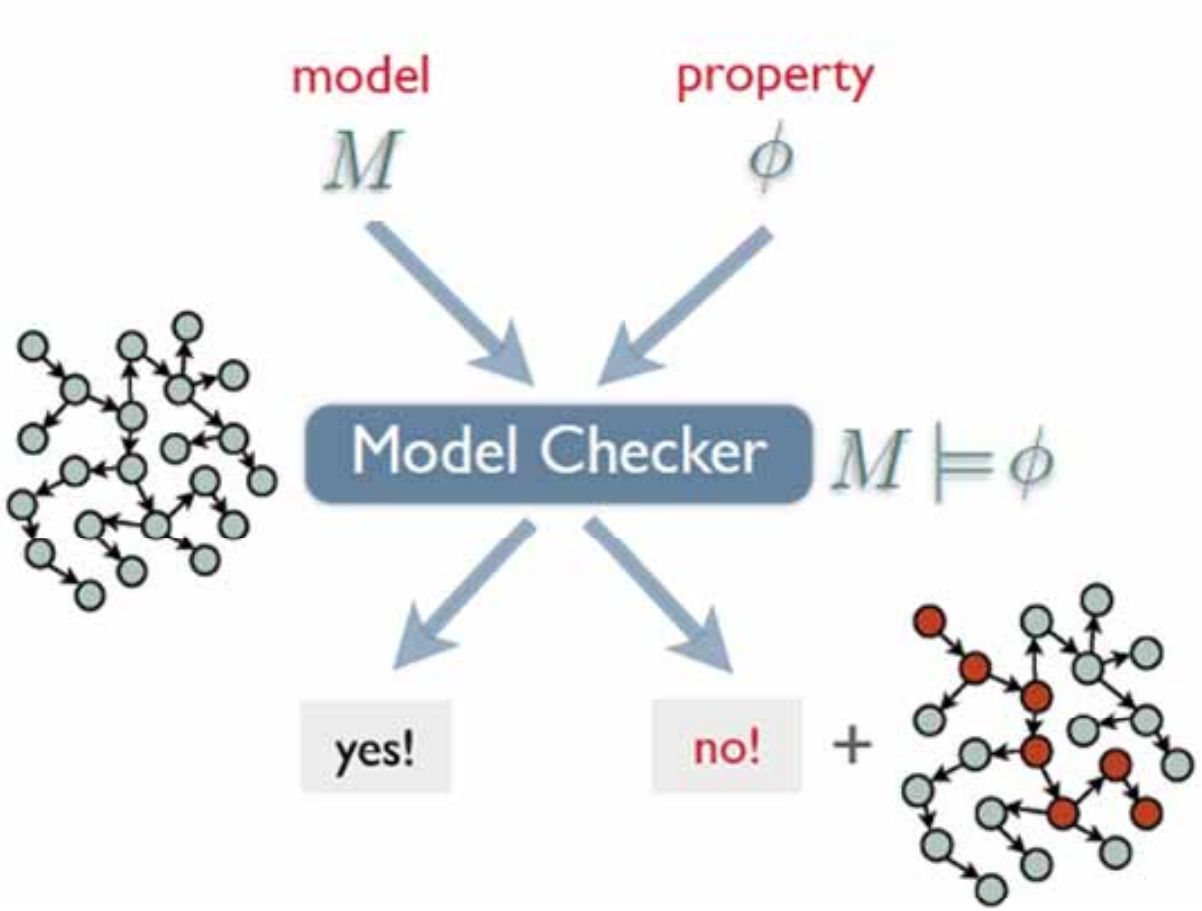
Verification of **Safety-Critical Systems** with **Formal Methods**

Introduction and Motivation II

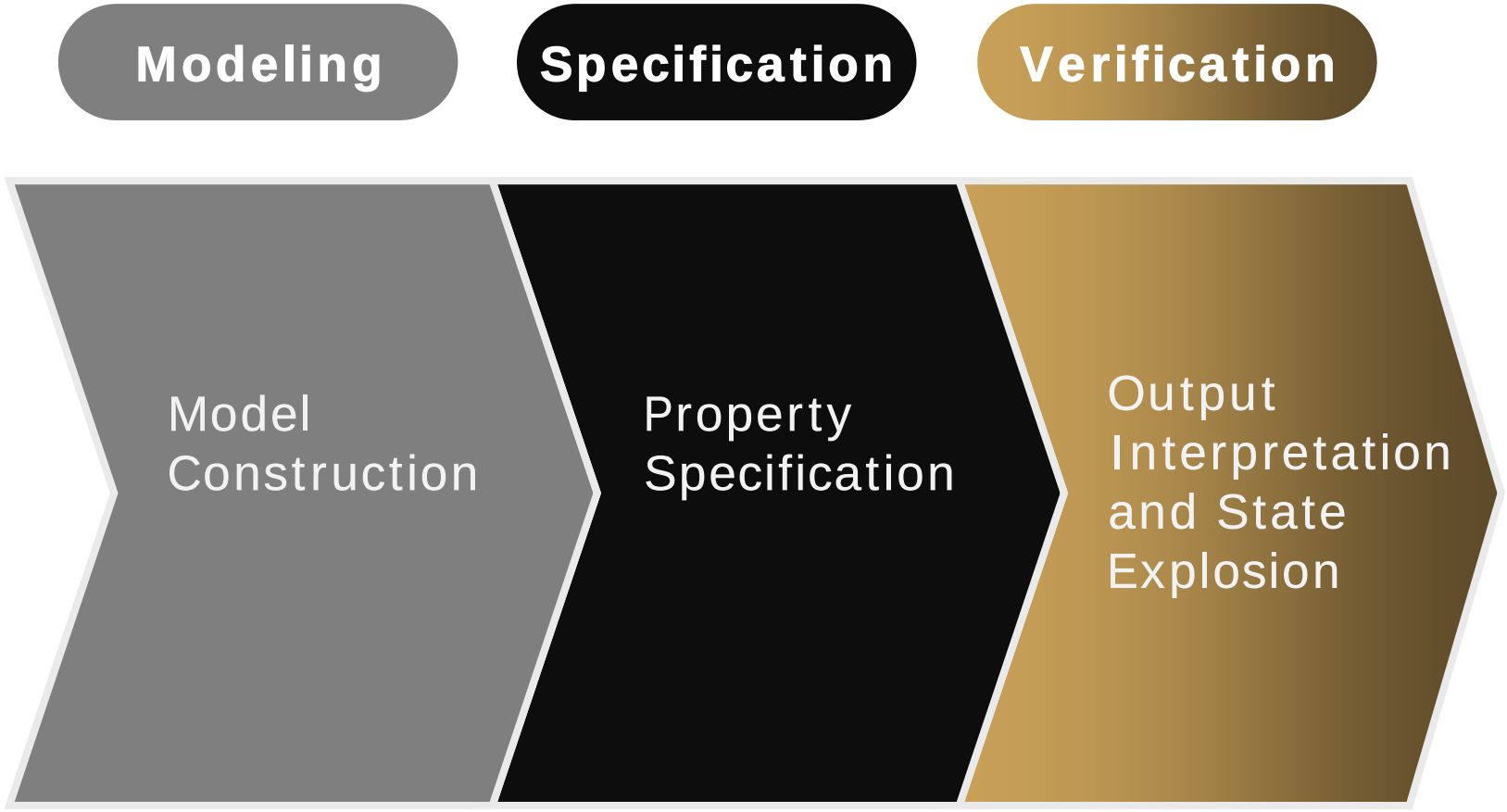
Verification of **Concurrent** Systems with **Model Checking**



Model Checking Technique



Model Checking Process



ATOS

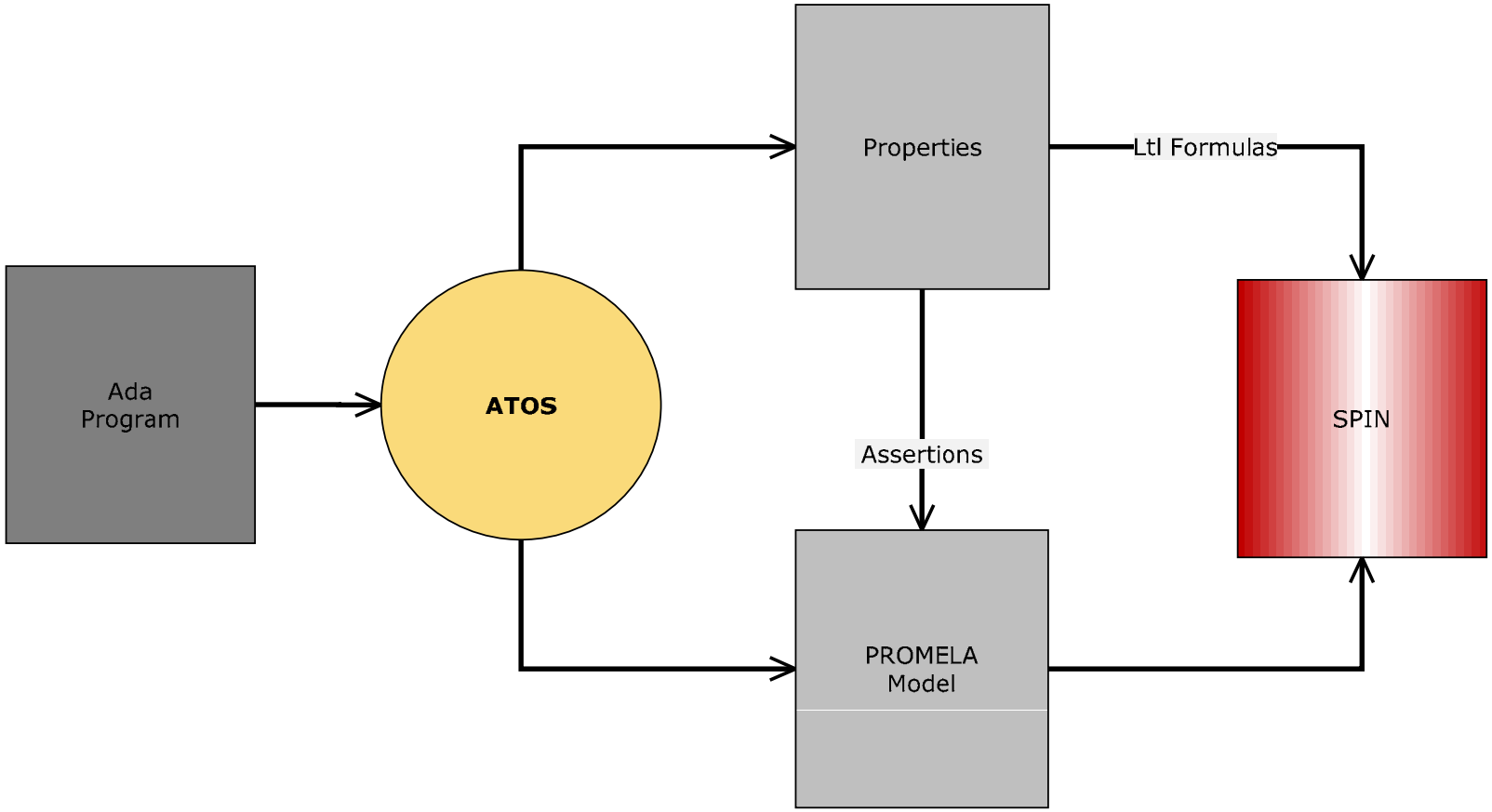
(**A**da translation

TO SPIN)

-

A Software Model Checker

ATOS I


Properties
LtI Formulas
Ada
Program
ATOS
Assertions
SPIN
PROMELA
Model

Why **Ada**?

Why **SPIN**?

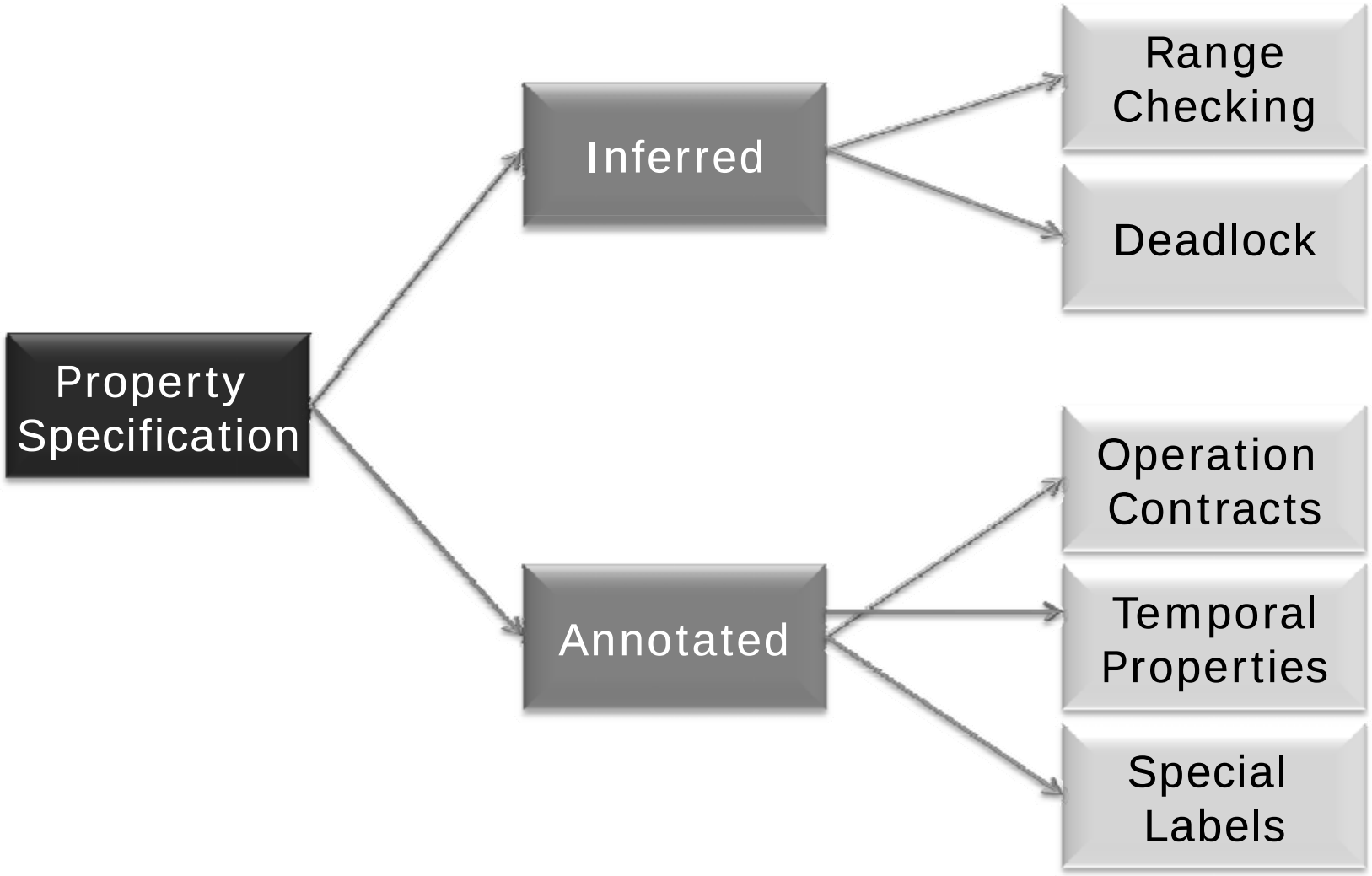
Model Extraction

Approach

Ada Covered Subset

Extending the Covered Subset

Property Specification



Inferred Properties

- Deadlock;
- Range Checking:
  - Extracts an LTL formula which checks if the range of a variable is violated;

[] (VarName **>=** lowerBound **&&** VarName **<=** UpperBound)

Annotated Properties

- An annotation language inspired by SPARK;
- Temporal Properties:
  - Pattern Properties;
  - User-defined properties;
- Operation contracts (procedures, functions and entries):
  - Pre and Post conditions (assertions);
  - Invariants (temporal property);
- Special Labels:
  - End;
  - Progress;
  - Accept

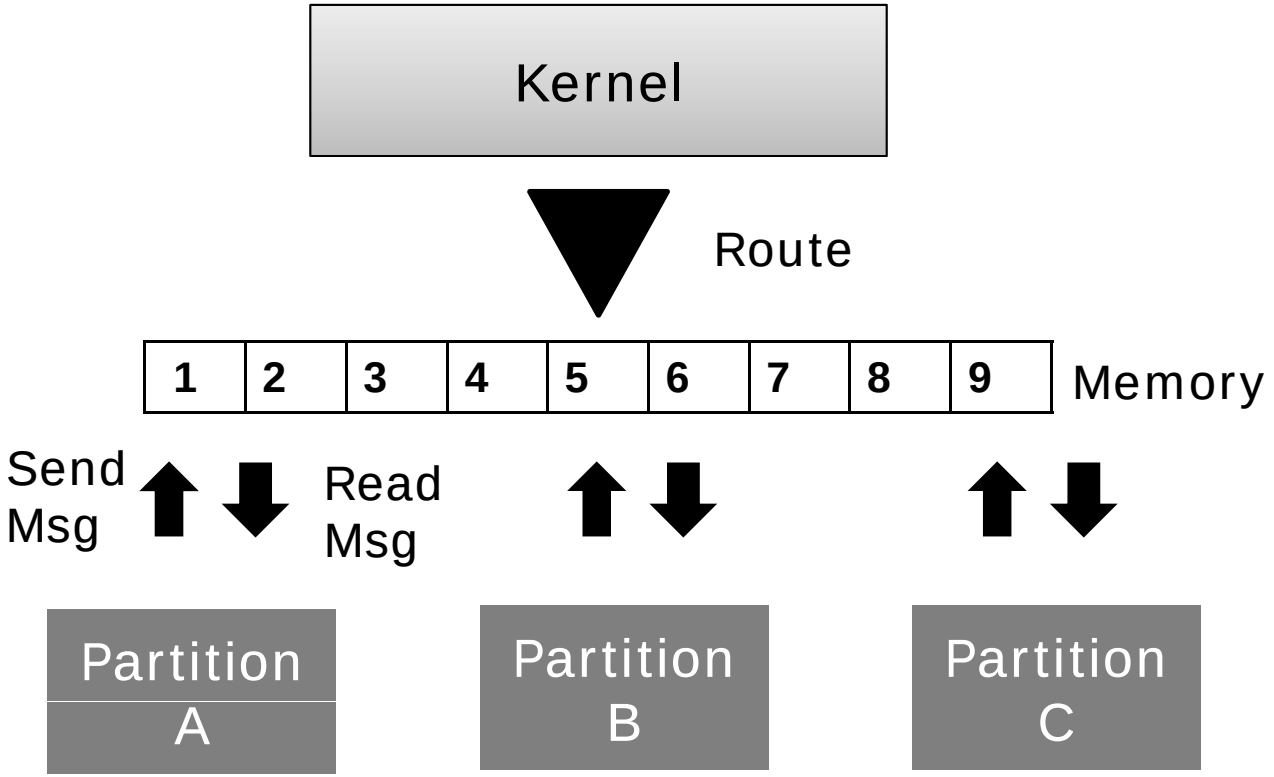
Experimental Validation - Separation Kernel

Policy

A			
B			
C			
	A	B	C

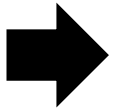
Pointers

A	1	2	3
B	4	5	6
C	7	8	9
	A	B	C



Model Extraction - Send Msg

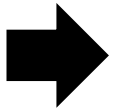
```
procedure Send_Msg(M: in Msg)
  is
    Origin: Proc_Id;
    Dest: Proc_Id;
 begin

    Origin := Get_Origin(M);
    Dest := Get_Dest(M);
    Write(M,
Pointers(Origin)(Dest));
    Flags(Origin)(Dest) := True;

 end Send_Msg;
```



```
inline UserTask0_Send_Msg ( M)
{

   UserTask0_Get_Origin ( M ,
UserTask0_Send_Msg_Origin );

   UserTask0_Get_Dest ( M ,
UserTask0_Send_Msg_Dest );

UserTask0_Write ( M ,
Mmr_Pointers [
UserTask0_Send_Msg_Origin + ( -1
) ].subPointers [
UserTask0_Send_Msg_Dest + ( -1 )
] );
}
```

Model Extraction - Kernel

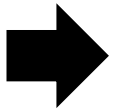
```
task KernelTask;

task body KernelTask is
     i: natural range 0..1:=
0;
   begin

     while i < 2 loop
         Route;
         i:=i+1;
     end loop;

   end KernelTask;
```



```
unsigned KernelTask_i : 2   =
0  ;

active proctype  KernelTask  (
) {

  do
    :: KernelTask_i < 2 ->
          KernelTask_Route (  );
          KernelTask_i  =
KernelTask_i + 1 ;
    :: else -> break;
  od
}
```

Model Extraction - Partitions

```
task type UserTask (TaskID:
Proc_Id);

task body UserTask is
      i: natural range 0..3 := 0;
      dest : Proc_Id;
      locMsg : Msg;
      A: Mem_Row;
   begin
      while i < 2 loop
           dest :=  (i mod 2) + 1;
           Read_Msgs(TaskID, A);
           Send_Msg(locMsg);
            i:=i+1;
     end loop;
   end UserTask;
```



```
proctype  UserTask0(  unsigned
TaskID:  2, byte ProcNumber) {

 do
   :: UserTask0_i < 2->
         UserTask0_dest=
               ( UserTask0_i % 2 ) + 1;

UserTask0_Read_Msg(TaskID,A);
         UserTask0_Send_Msg
           (UserTask0_locMsg  );
         UserTask0_i  =  UserTask0_i
+ 1  ;
     :: else -> break;
od;
```

Property Specification I

Once initialized the information flow policy is immutable

```
--# property Policy_Space(1)(2) and not Policy_Space(x)(y)
--# is_true after main@PolicyEnd
```

```
[] (main@PolicyEnd -> [] ( Policy_Space[1][2] && not
Policy_Space[x][y] )
```

Property Specification II

The memory space of a partition can only contain messages whose sender is authorized

```
--# property (Flags(x)(y) and
--# Mem_Space(Pointers(x)(y)).Dest=x
--# and Mem_Space(Pointers(x)(y)).Origin=y ) ->
--# Policy_Space(y)(x) is_true globally
```

Conclusions and Future Work

- ATOS potential?

- The Ada scheduler is not represented in SPIN. What are the consequences?

- The **Output Interpretation** and **State Explosion** problems.

Informatics Department
University of Minho

Ada Europe

2012

An Approach to Model Checking Ada Programs

Authors: João Martins, José Miguel Faria and Jorge Sousa Pinto

Stockholm, 13th June, 2012