TRASYS

WE GET IT DONE

RCOS – Main Entities

Konstantinos Kapellos, Roger-Pissard Gibollet – TRASYS
Andrea Merlo – Thales Alenia Space - Italy

May 2015, ESTEC

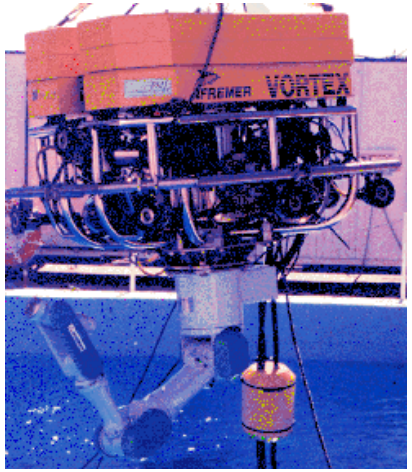
Main drivers

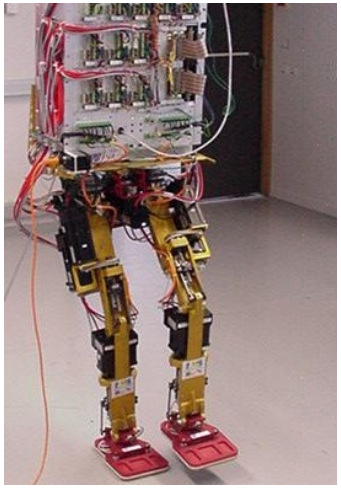
- In general, physical tasks to be achieved by robots can be stated as *automatic control problems* which can be efficiently solved in real-time by using adequate feedback control loops.
- The characterisation of the physical task is not sufficient for fully defining a robotic action: *starting and stopping times* must be considered, as well as *reactions to significant events* observed during the task execution.
- The overall performance of the system relies on the existence of *efficient real-time mechanisms* at execution level.
- For the design of a robotic application are needed formalisms supporting *sequencing*, *parallelism, synchronisation, preemption, …*

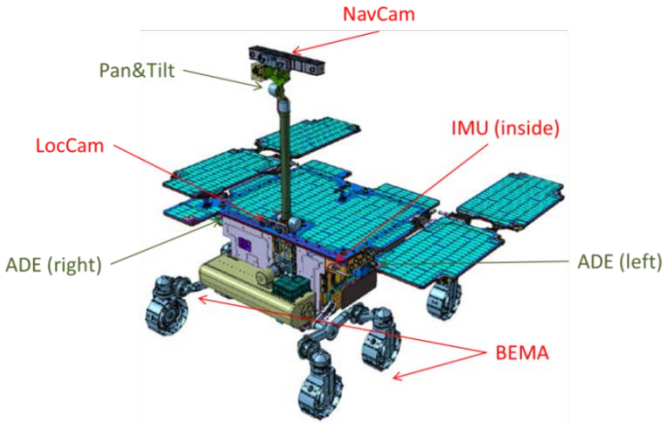
- Users
  - Control system engineer: to design and program control laws
  - End user: specification of complex robotic applications in a safe way
- Tools
  - Automatic code generation

TRASYS
WE GET IT DONE

Examples of targeted applications





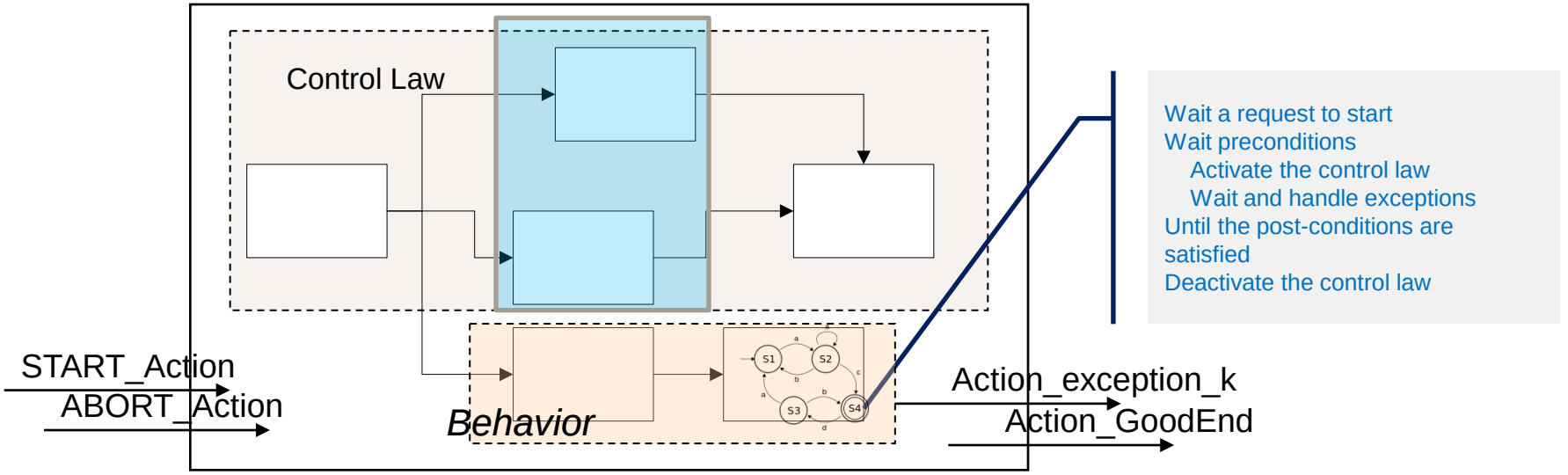


ExoMars rover Activities: used to illustrate the methodology

TRASYS
WE GET IT DONE

Main entities: Actions and Tasks

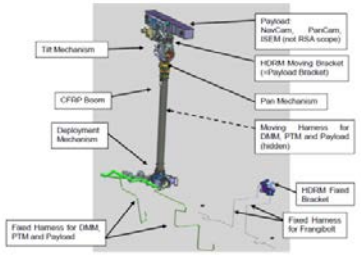
- Action is defined as the parameterized specification of:
  - A control law structurally invariant along the Action duration
  - A logical behavior associated with a set of events which may occur just before, during and just after the task execution (pre-conditions, exceptions, post-conditions)
  - A set of temporal constraints (grouping of modules in real-time tasks, sampling period, estimated execution duration, …)



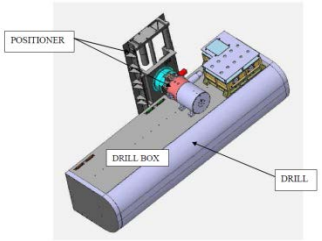
TRASYS
WE GET IT DONE

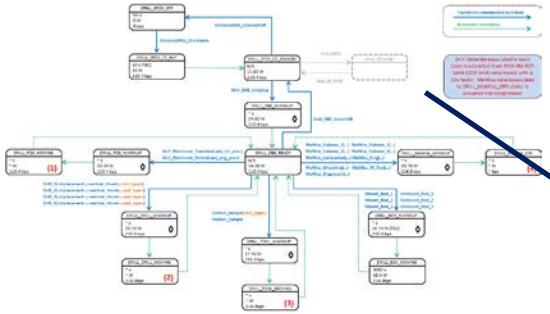
Example from the ExoMars mission

- A set of ~200 Actions are defined for all the subsystems

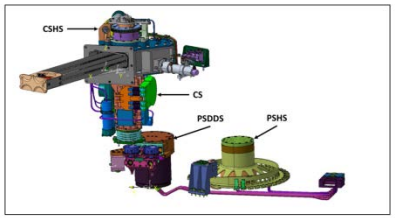


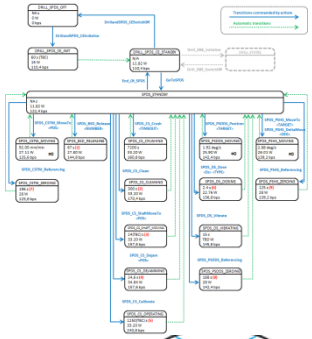
MAST_PTU_Initialise
MAST_PTU_MoveTo (params)
MAST_DEP_Initialise
MAST_DEP_MoveTo
MAST_SwitchOff





DrillandSPDS_CEInitialise
Drill_DBE_Initialise
Drill_Positioner_Translate (params)
Drill_Positioner_Rotate (params)
Drill_1 (params)
Collect_Sample
Mount_Rod_1
…





…
SPDS_CSTM_MoveTo (params)
SPDS_CSTM_Referencing
SPDS_CS_Crush
SPDS_CS_Clean
SPDS_PSDDS_Position (params)
…

TRASYS
WE GET IT DONE

Main entities: Actions and Tasks

- Task is defined as the parameterized specification of:
  - a *main body*, (nominal execution of the Task), *composed of Actions, Tasks and conditions which fulfils the goal of the Task*
  - *recovery bodies* to handle the exceptions
  - a pre-defined behaviour for the logical co-ordination of the previous items: the main body of the Task is activated after satisfaction of the pre-conditions, and normally ends when the post-conditions are satisfied. If an exception occurs, the nominal execution is aborted and replaced by the specified recovery body.

- The composition of Actions and Tasks in the main and the recovery bodies is obtained through operators which express sequence, parallelism, conditions, iterations, rendezvous, usual control operators and various levels of pre-emption.

- Use of domain specific languages relying on the semantics of formal languages

Example from the ExoMars mission

- ~60 Tasks are defined
  - RV_Prepare4Travel

```
var with_wisdom: integer in
 loop
   [ %
       await Task_RV_Prepare4Travel_start;
        [
            [
               emit A_ADE_Left_Initialise_START;
              ||
               emit  A_ADE_Right_Initialise_START
            ]
            ;
            [
               await A_ADE_Left_Initialise_GoodEnd
              ||
               await A_ADE_Right_Initialise_GoodEnd
            ]
            ;
            [
               emit A_DMA_Initialise_START;
              ||
               emit A_GNC_Initialise_START;
             ]
             ;
             ……..
              ;
              call DeactivateTask()("RV_Prepare4Travel");
    ]
    end loop
    end var
   ]
  ||
```

  - Programmed using Esterel

TRASYS
WE GET IT DONE

Formal Verification

- Structured specification allows to automatise the formal verification

  – Critical properties

  – Relationship between Actions/Events and Actions

  The execution of the Action/Task A1/T1

  … triggers …

  … is always preceded by …

  … always takes place during …

  … always implies …

  the execution of the Action/Task A2/T2

  The occurrence of the Event event

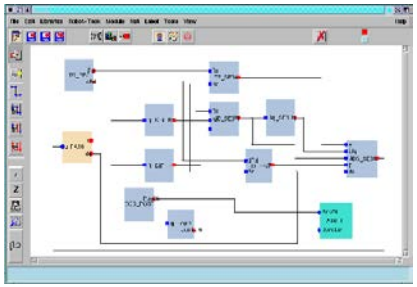
  … triggers …

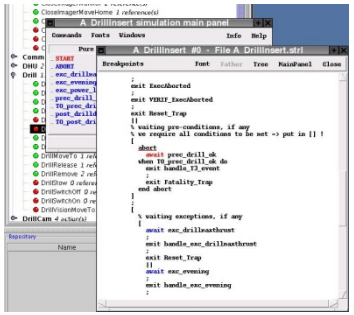
  the execution of the Action/Task A2/T2

TRASYS
WE GET IT DONE

Tools

- ORCCAD Tools (INRIA Sophia-Antipolis, B. Espiau)



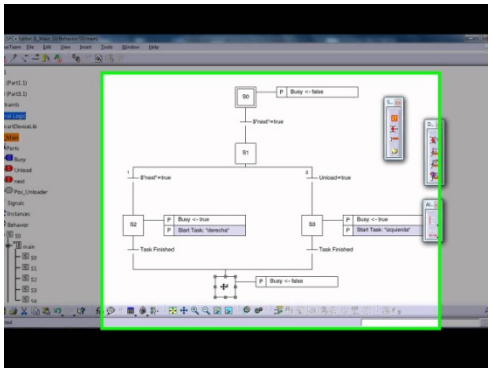


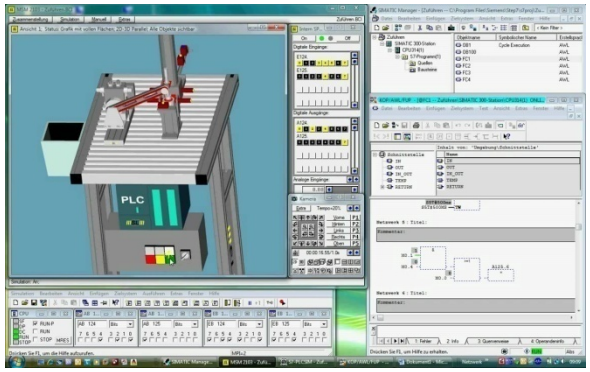
Ilog Views

Esterel

Execution Real-Time lib

- DELMIA Automation (DELMIA V5 - CLM Control Setup)





- ESA Activities
  - VIMANCO, MUROCO, …., IRONCAP, …, 3DROV/ROSEX

TRASYS

WE GET IT DONE

Conclusion

- The proposed approach considers as main entities:
  - The **Modules**: algorithms that implement the necessary functions of the targeted system
  - The **Module Tasks**: the real-time contexts in which the Modules are running in a periodic/sporadic, possibly multi-rate way
  - The **Actions**: hybrid elementary entities encapsulating the continuous time aspects into a Discrete Event behavior
  - The **Tasks:** parameterized logical and temporal compositions of Actions and Tasks using sequence, parallelism, conditions, iterations, rendezvous, pre-emption, ...