Verification and Validation meet Planning and Scheduling

AndreA Orlandini

(CNR-ISTC)

Email: **andrea.orlandini@istc.cnr.it**

National Research Council of Italy (CNR-ISTC)

P&S Autonomy and V&V

- P&S systems are finding increased application in mission safety-critical and dependable systems

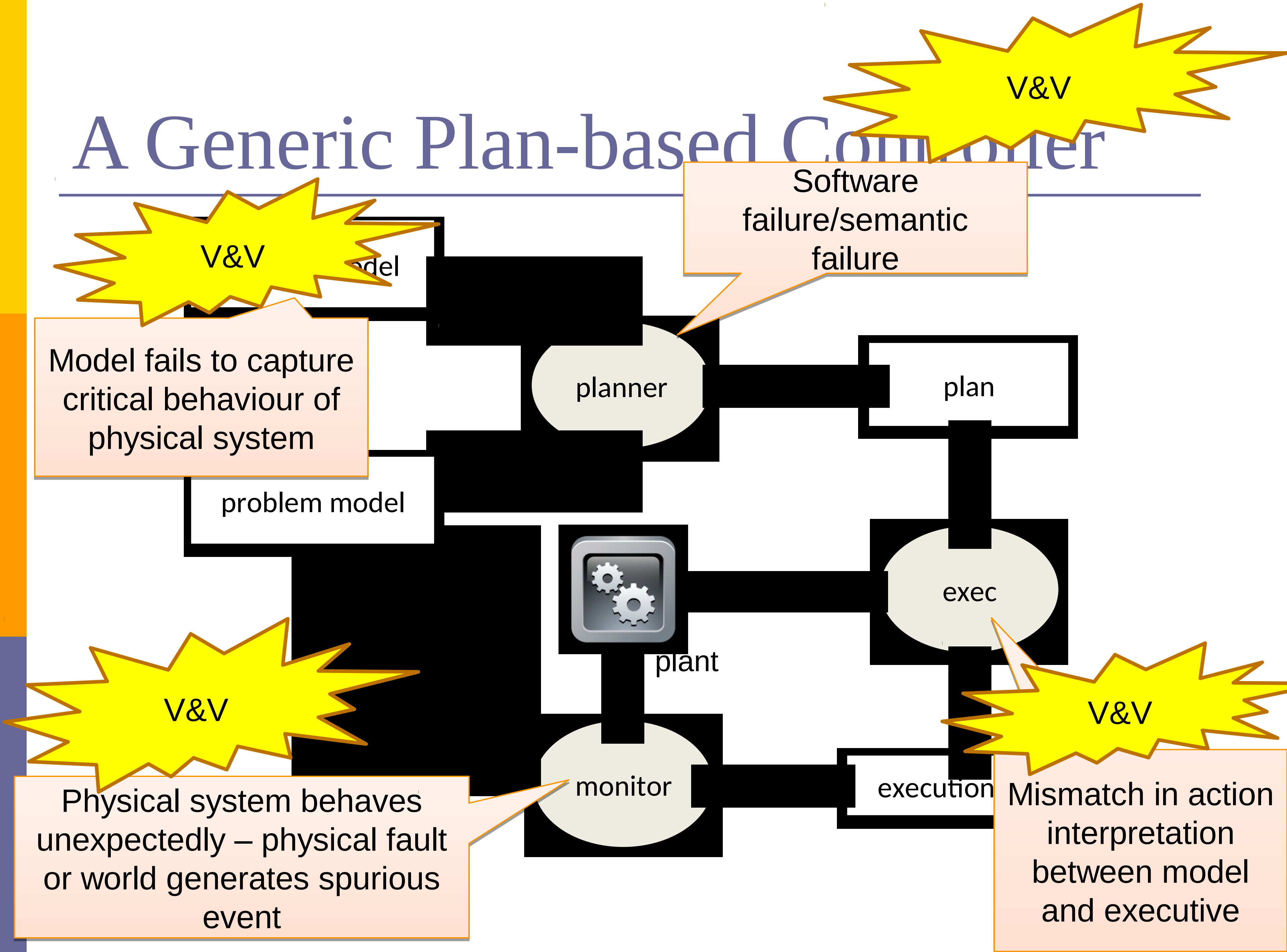
- Model-based autonomy to generate plans to control a *plant*, e.g., a spacecraft or a rover

- A first relevant example in a real-world context
  - Remote Agent Experiment (RAX) for Deep-space 1 (DS-1) mission endowed with V&V technology – Livingstone (2001)

(Jonsson et al. AIPS 2001)

- Nevertheless, tools and methodologies for V&V of P&S have received relatively little attention…

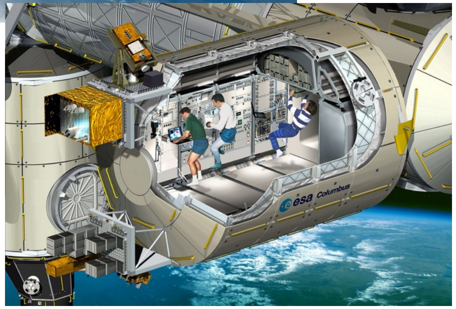
A Generic Plan-based Controller



ULISSE, USOCs and Increment Planning Processes

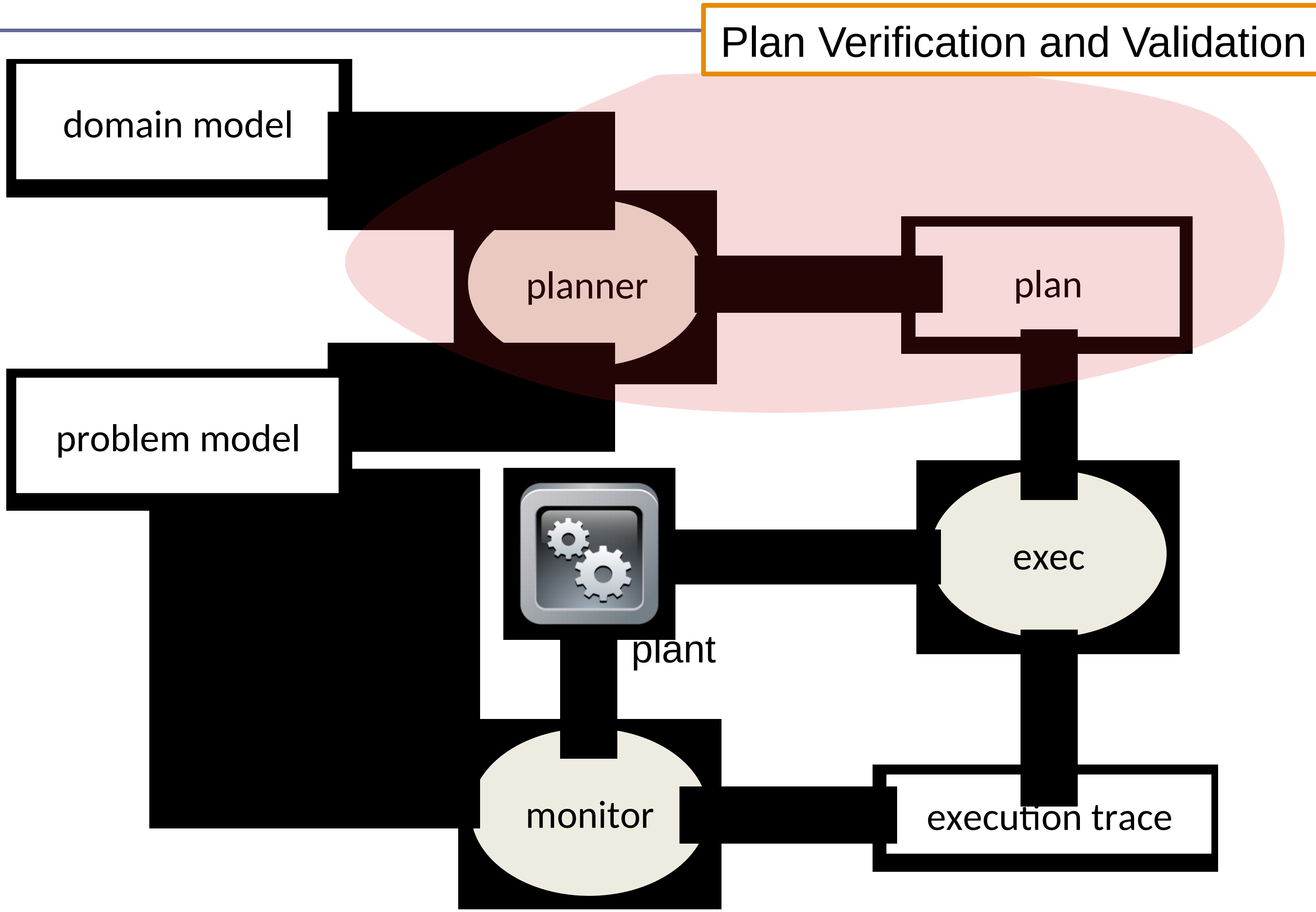
- *ULISSE* aims at improving *preservation, valorization and exploitation* of data produced by the *Columbus* module on board the *ISS*

- *USOCs* are a network of scientific space facility operations centres



- To produce a demonstrator we have targeted the *ISS increment planning* (usually a three months period)

A Generic Plan-based Controller

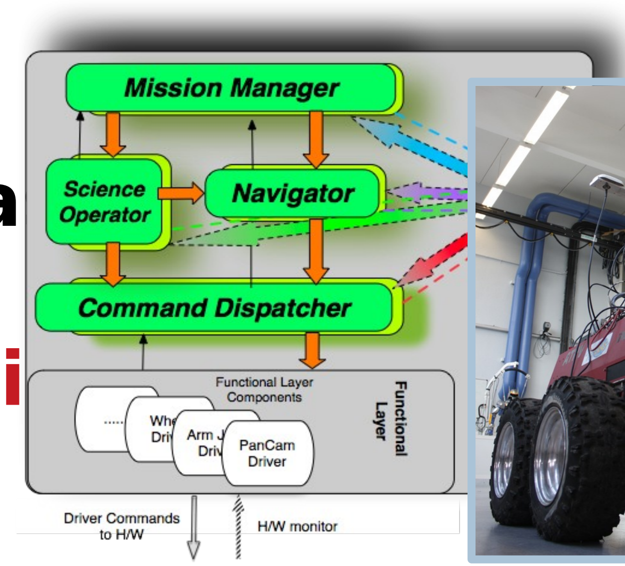


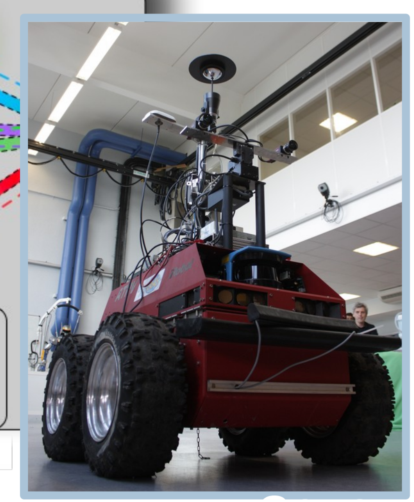
Goal Oriented Autonomous Controller (ESA ITT [2009-2011])

- Aiming at creating a **state of the art autonomous controller** for ESA's space rovers

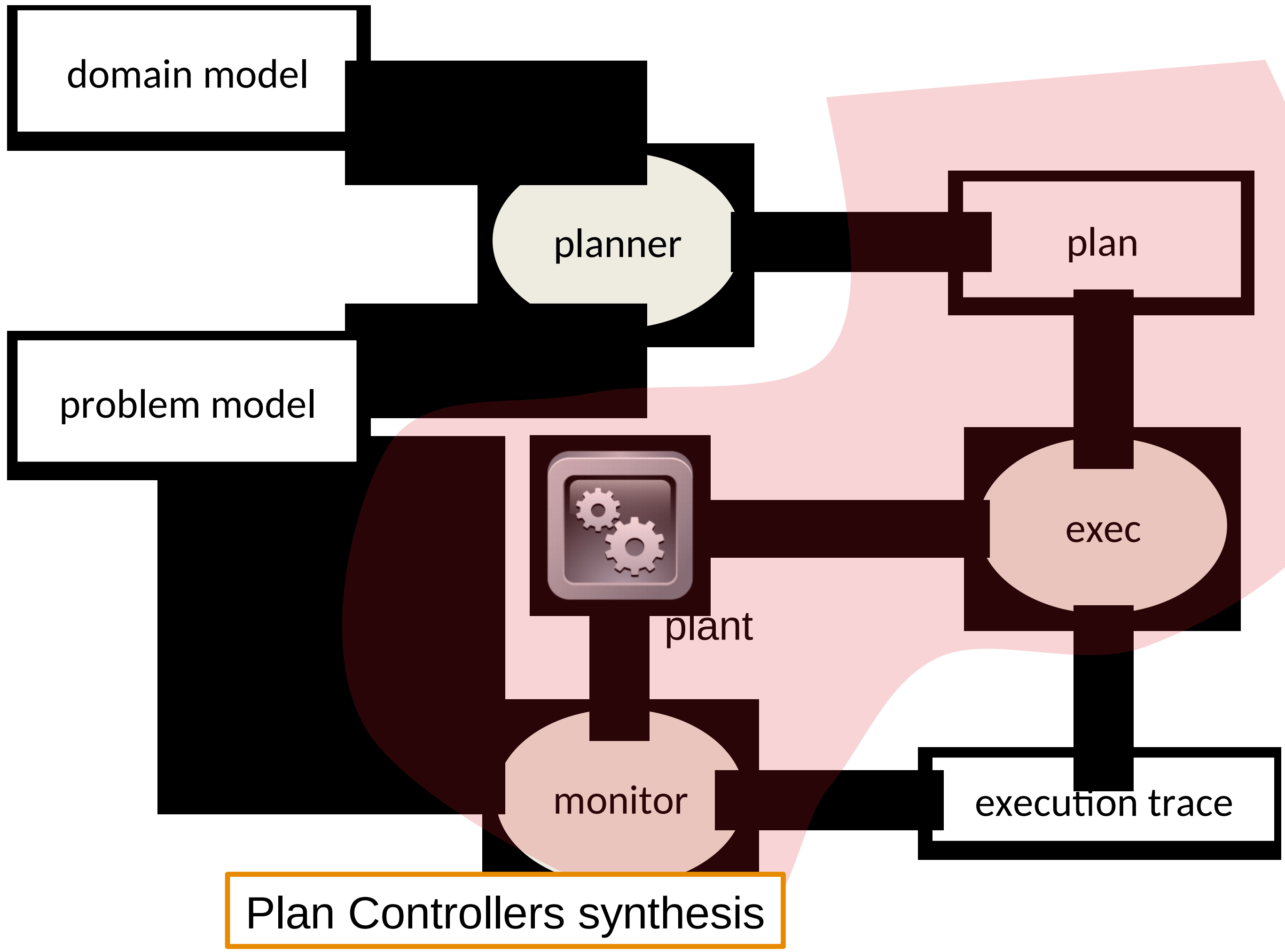
- Consortium: GMV (Spain), LAAS (France), Verimag (France), MBARI (USA), CNR-ISTC (Italy)

- **Deliberative layer ba on timeline-based planning and executi**





A Generic Plan-based Controller

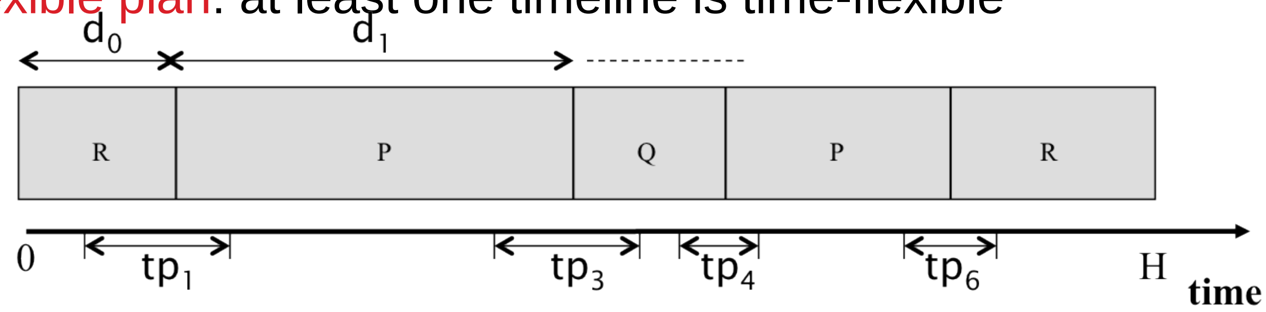


Timeline-based Planning

- Plans are set of *Timelines*
  - A timeline denotes the temporal evolution of a particular feature (*State Variable*)

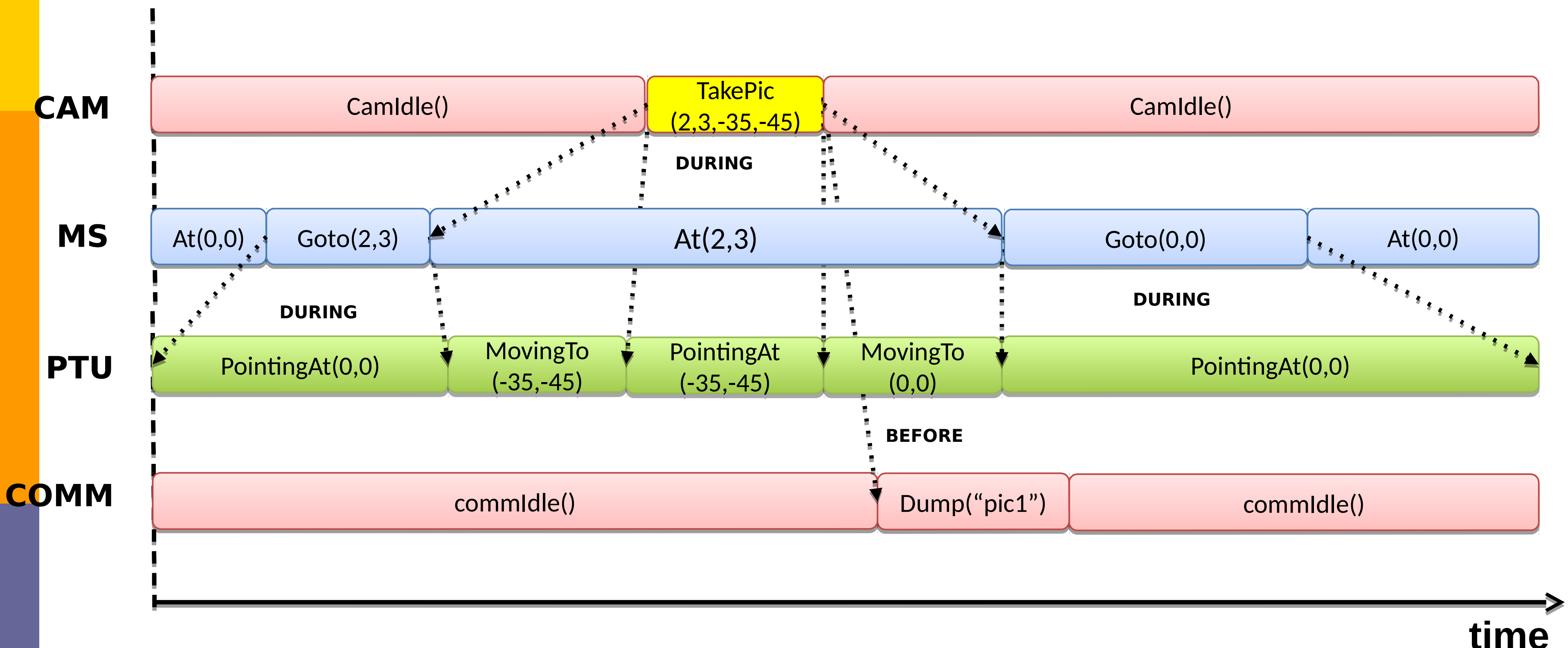
- A *Domain Theory* describes a planning domain defined over a set of State Variables by means of *Synchronizations*

- Time-flexible plan: at least one timeline is time-flexible



- Planning process should build a *valid plan* (w.r.t. a Domain Theory) achieving the desired Goals

Timeline-based solution plans



Timelines Execution

- An plan executive cannot completely predict the behavior of the controlled system
- A *Controllability Problem* can be defined distinguishing between *contingent* and *executable* processes [Vidal and Fargier 1999]
- The *Dynamic Controllability* definition has been extended to Timelines [Cesta et al. 2009]
- A suitable Plan Controller is required

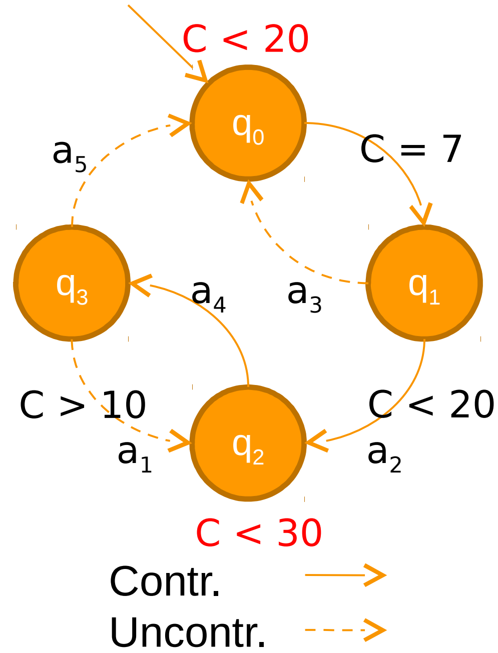
$$\mathcal{C} : \mathcal{PB} \times \mathbb{N} \rightarrow \mathcal{V}_1 \cup \ldots \cup \mathcal{V}_n \cup \{\lambda\}$$

(with PB the set of partial behaviors defined by a time flexible plan over a partial horizon H’ < H
to the set of controllable values or wait action)

Timed Game Automata

[Maler & Pnueli & Sifakis 1995]

- Act is split in two disjoint sets
  - Act_c : the set of controllable actions
  - Act_u : the set of uncontrollable actions
- A *valuation* is a mapping from the set of clocks to integers
- A *state* is a pair (q_i,*v*) with *v* a valuation
- A *strategy* *F* is a partial mapping from the set of Runs of *A* to the set of $Act_c \cup \{\lambda\}$
- The special action λ stands for "just wait and do nothing"



Building TGA from Timelines

- Verifying flexible plans solving TGA Reachability Games

- Encoding into an adequate set P*L* of TGA:
  - Flexible Plan
  - State Variables
  - Domain Theory

- Suitably defining a TGA Reachability Game (RG)
  - Winning the game implies Verifying the plan (and checking DC)

- UPPAAL-TIGA as a verification engine

[Cesta et al. 2009, Cesta et al. 2010]

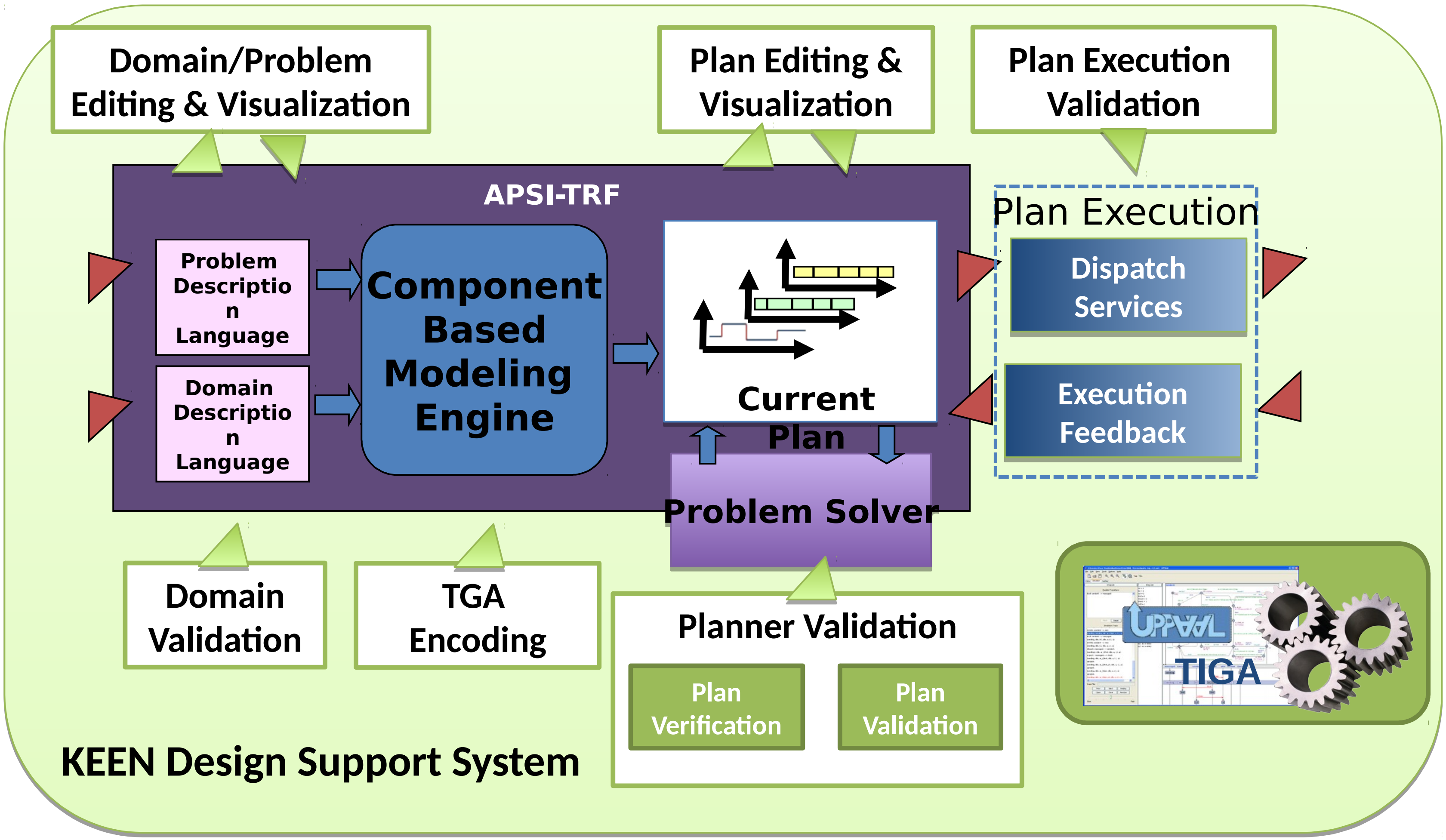
Synthesizing Controllers

- For each partial behavior *pb* in *PB*, it there exists a unique related run r_{pb} of *PL*

- Definition 2. Given a suitable *RG defined on PL*, the winning strategy *f* generated by UPPAAL-TIGA defines a plan controller C_f as follows:
  - for each *pb* in *PB* over H' $C_f(pb,H') = f(r_{pb})$
  - otherwise $C_f(pb,H')$ is undefined

[Orlandini et al 2011, Orlandini et al 2013]

APSI-TRF and Knowledge Engineering ENvironment (KEEN)



[Orlandini et al 2014]

RESTART!!!

Current Work: (Re)Formalization of Flexible Timelines and its Execution

- Formal characterization of flexible timelines and plans still missing

- Formal definition of
  - Planning Domains and Goals
  - Tokens/Timelines, Flexible Tokens/Timelines and Plans

[Cialdea Mayer et al 2014]

- Revised definitions
  - Difference between ***controllable*** and ***uncontrollable*** activities
  - Quantitative temporal relations
  - Execution semantics in terms of TGA

[Cialdea Mayer & Orlandini 2015]

Flexible tokens, timelines and plans

- A ***flexible token*** (v,[e,e’][d,d’],γ) is a valued interval characterized by a value *v*, end [e,e’] and duration [d,d’] intervals and a controllability tag (*c,u*)
- A ***timeline TL*** is a sequence of flexible tokens
- A ***set of timelines FTL*** describes a possible (temporal) evolution of a system
- ***(Allen’s) Temporal relations***
  - Between intervals
  - Between interval and a timepoint
- A ***flexible plan is a pair (FTL, R)***

Situations and Execution Strategy

- Given a set of timelines FTL, a ***situation*** ω is a function to assign values to uncontrollable tokens (Ω set of situations)

- ω(FTL) defines a ***projection*** of FTL – i.e. a fully controllable evolution of FTL

- A ***scheduling function*** θ assign an execution time to every controllable token (T set of scheduling functions)

- An ***execution strategy*** for a flexible plan is a mapping σ : Ω_{FTL} 🖧 T_{FTL}

Execution strategy and Controllability

- An execution strategy is viable when ***consistently*** applied to the plan

- A plan may be
  - ***weakly controllable*** – there is a viable execution strategy for each situation
  - ***Strongly controllable*** – there is a viable execution for every situation
  - ***Dynamically controllable*** – there is an *dynamic execution strategy* (DES) for all situations – decisions only considering past uncontrollable events

Current Work(2): A more comprehensive approach

- Comprehensive formalization of timeline-based planning and its execution
- Controllability information included in the domain/plan description
- TGA encoding does not require to consider also the planning domain specification
- More compact and straightforward translation of plans in terms of TGA
- Possibility to encode also partially specified plans

Future Works

- Deployment in a real P&S framework
  - E.g. APSI-TRF framework
  - **STNU-based representation independent!!!**
- Extension of formalization to consider also resources and scheduling
- More tight integration of planning and verification
  - Verification tool to check partial plan consistency