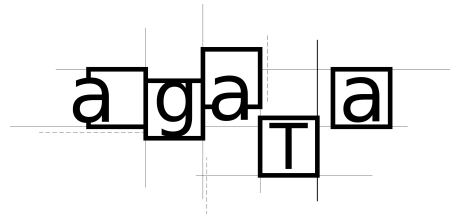


Interaction between reactive and deliberative tasks for on-line decision making

Michel Lemaître, Gérard Verfaillie
ONERA, Toulouse, France



Context of this work



Objective: to develop techniques for increasing spacecraft autonomy.

Example of scenario: a constellation of autonomous satellites dedicated to detection and observation of ground phenomena.

Preliminaries

Please, **forget** everything you know, everything you are used to about the usual **three level** robotic architectures: planning, executive, and real-time execution!

Basic requirements for autonomy

Need for a **reactive control**.

Examples: how to **react instantaneously** to a ground phenomenon detection, to an observation failure, or to an unexpected level of energy.

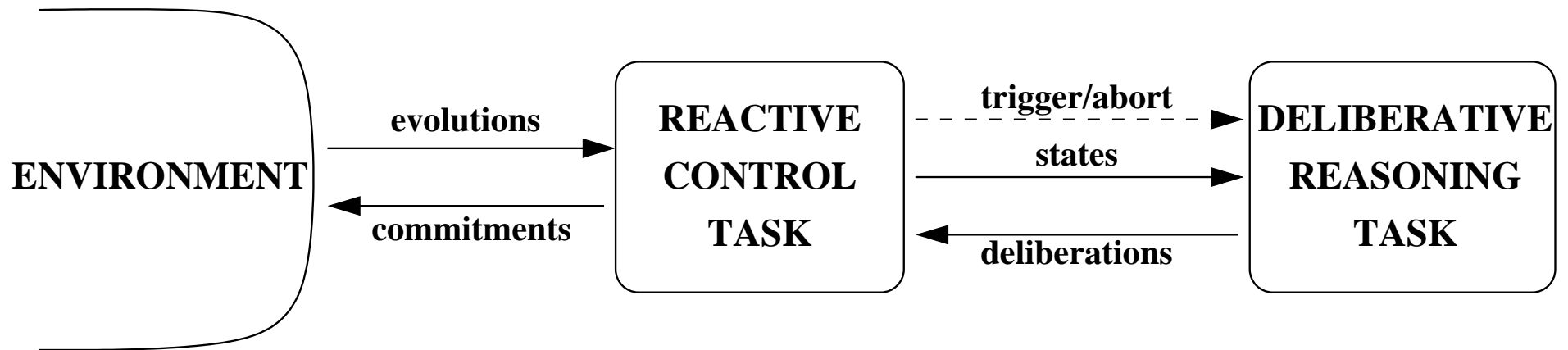
Need for **deliberative reasonings** for **on-line decision-making**.

Example: to build an observation and data downloading plan over a given temporal horizon ahead, in order to decide upon the best action to perform next.

Cannot be **instantaneous**.

Question: which **interaction** between them?

A generic interaction scheme



The deliberative reasoning

Anytime behaviour.

Ability to **produce quickly** (not necessarily instantaneously) a first result and to **improve on it** as long as computing time is available.

No **hard** temporal requirements.

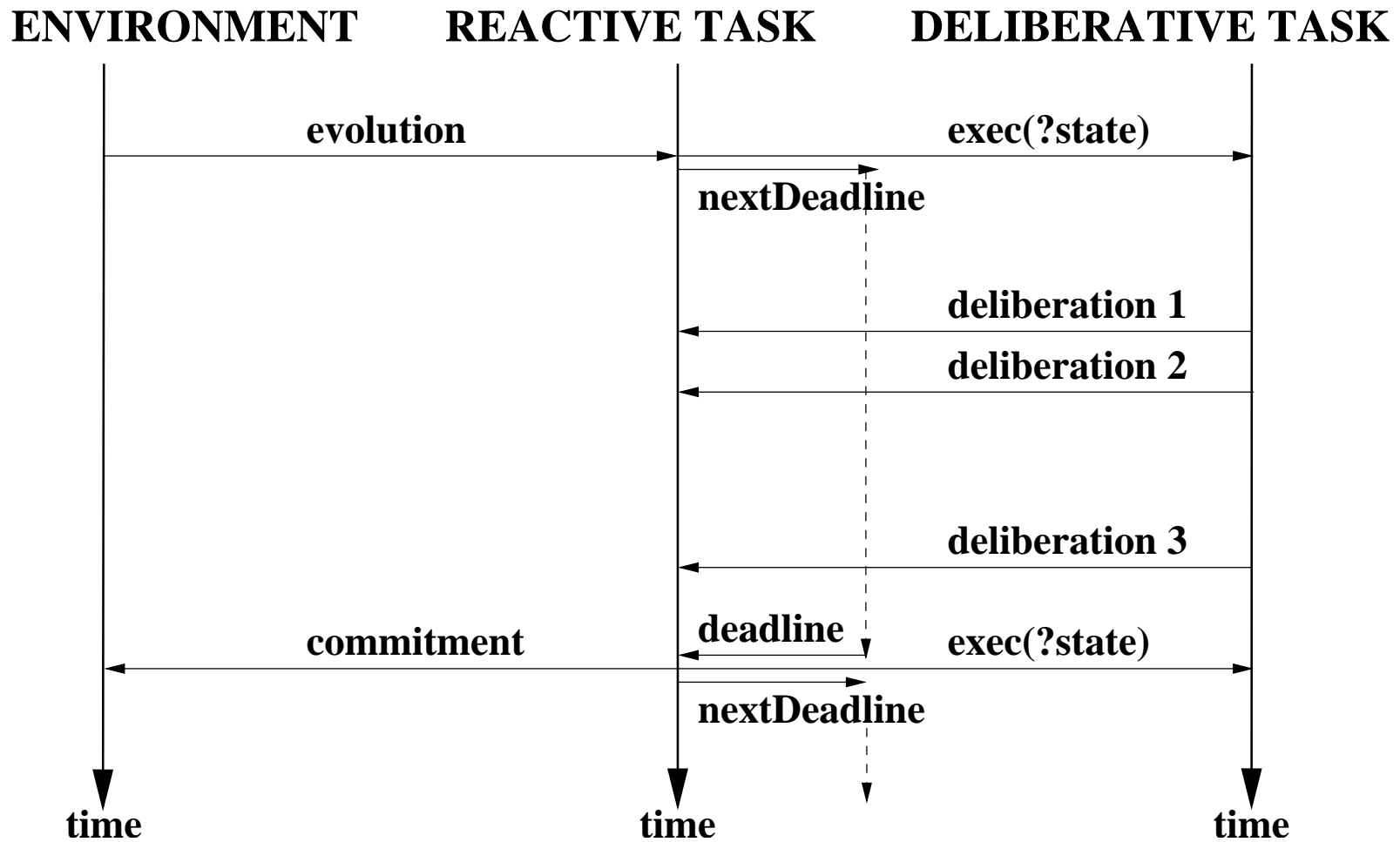
The reactive control

Ability:

1. to **filter evolutions** from the environment;
2. to **compute deadlines** for decision-making;
3. to **trigger deliberative tasks** and to provide them with relevant information about the state of the environment and of the objectives;
4. to **check deliberations** before decision-making when deadlines occur;
5. to **compute default decisions** when no decision is available when deadlines occur.

Everything abstractly **instantaneously**,
that is in fact before the next evolution.

Example of temporal execution



Implementation

1. the **synchronous language Esterel** for the **reactive control**;
2. the **standard language Java** for the **deliberative reasoning**;
3. the **Esterel “Task” mechanism**, which allows any external task to be triggered, synchronized, and aborted from an Esterel program.

Main **Esterel** simplified loop:

```
1 every evolution do
2   loop
3     exec deliberative(?state)
4     return deliberation ;
5   end loop
6 end every
```

Everything **compiled** into a **Java code**.

Experimentation



Simulation on the scenario of an autonomous satellite dedicated to detection and observation of ground phenomena.

That works !