A generic architectural framework for the closed-loop control of a system

Gérard Verfaillie, Michel Lemaître
ONERA, Toulouse
Marie-Claire Charmeau
CNES, Toulouse

ONERA
THE FRENCH AEROSPACE LAB

cnes

The AGATA project

AGATA =
Autonomy Generic Architecture: Tests and Applications.

Joint project between **CNES**, **ONERA**, and **LAAS-CNRS**, in Toulouse.

Start : mid-2004. **End of first phase**: mid-2008.

Manpower : 3-4 engineers / year / organization.

Information : http://agata.cnes.fr

The AGATA project

Long-term objective: to increase **spacecraft autonomy**, in order to cast off the **constraints on communications** between Earth and the spacecraft:

1. visibility **windows**,
2. **incompatibility** between communication and normal mission execution,
3. communication **time**,

and thus to improve **system reactivity** and **dependability**, and **mission return** in terms of quantity, quality, and quick delivery of collected data.

Medium-term objective: to design, to implement, and to experiment a **ground demonstrator** of an autonomous satellite.

Example

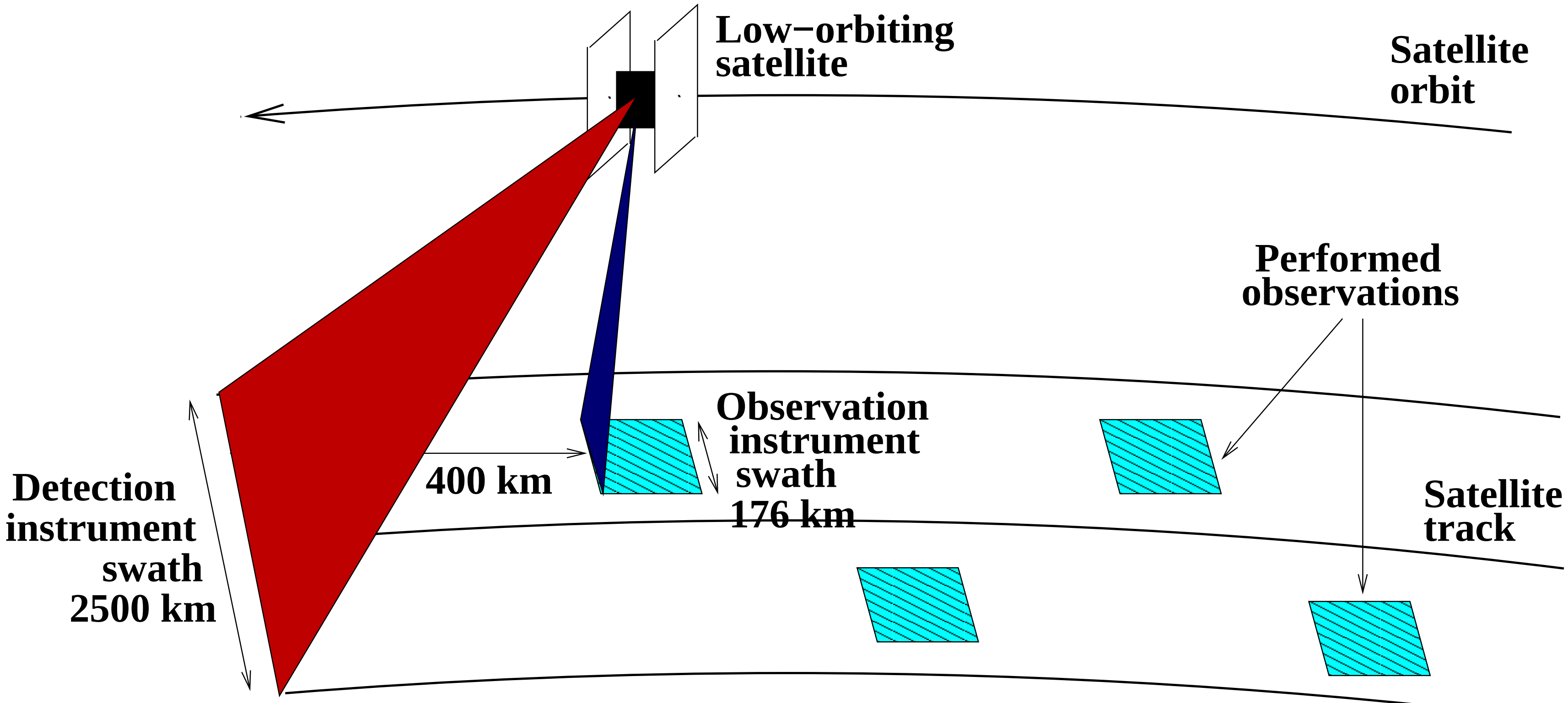
An **Earth-orbiting** satellite equipped with:

1. a wide-swath **detection** instrument in front of the satellite,
   able to detect hot spots due to forest fires or volcanic eruptions;
2. a narrow-swath orientable **observation** instrument,
   able to make observations of specific ground areas;
3. a mass **memory**
   to record observation data;
4. a high-rate **antenna**
   to download them when in visibility of a ground station.

Constraints on **communications**: visibility of a ground station only 10% of time.

Useful **onboard decisions**: observation + data recording and downloading.

Earth surveillance and observation satellite



A generic architectural framework

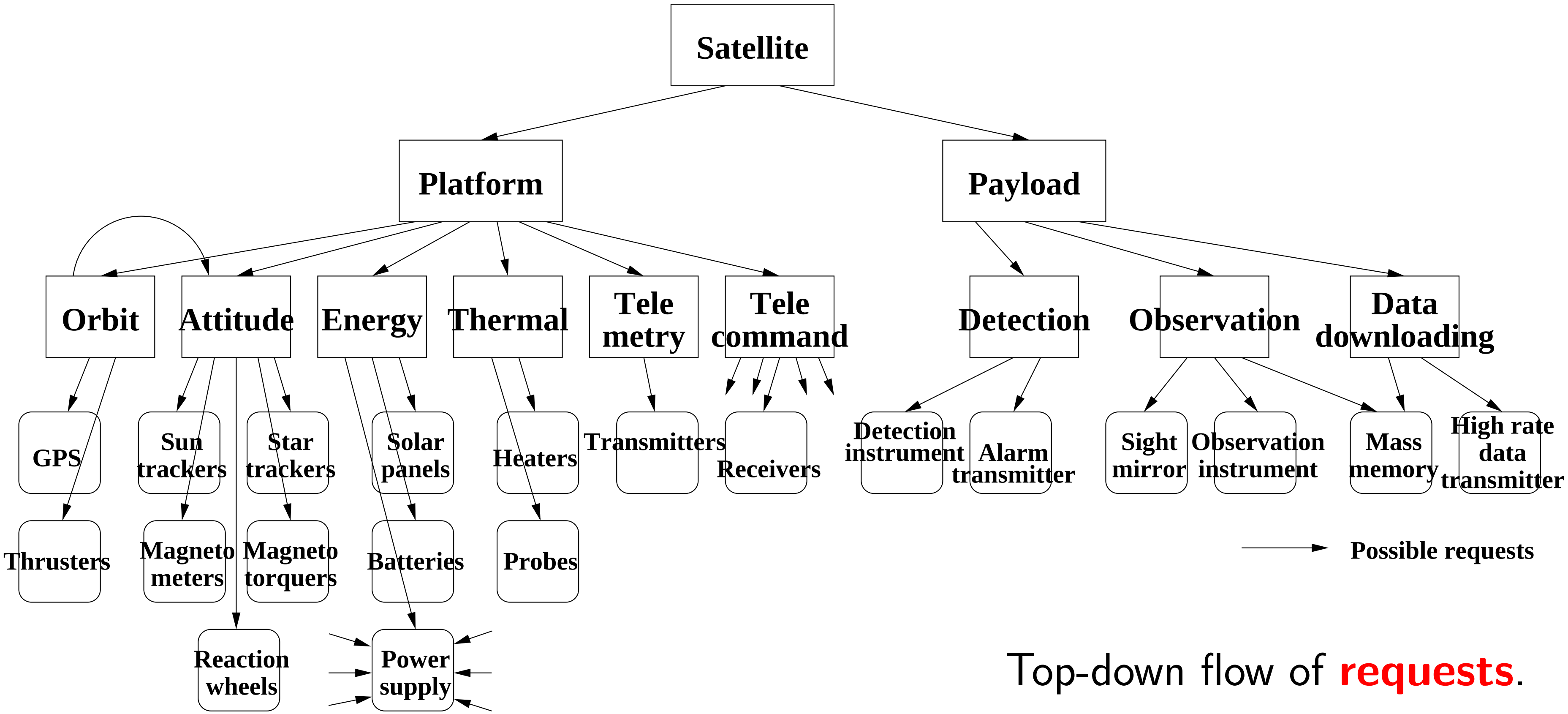
Definition of a generic architectural framework for the **control** of an autonomous satellite: one of the first tasks of the AGATA project.

Main features:

1. control **decomposition** into hierarchically organized modules;
2. control and data **encapsulation** in each module;
3. standardized **communications** between modules;
4. generic **organization** of each module;
5. generic scheme of **interaction** between **reactive** and **deliberative** tasks.

Control decomposition into hierarchically organized modules

A possible decomposition for an Earth **surveillance** and **observation** satellite.



Top-down flow of **requests**.

Bottom-up flow of **information**.

Possible **dependencies** between two modules taken into account at a higher level.

Encapsulation of control and data in each module

Each **module** in charge of the control of a **subsystem**.

If the module M is responsible for the control of the subsystem S:

1. S cannot be controlled from any other control module M' without a **request** to M;

2. information about the state of S cannot be obtained without an **access** to the data maintained by M.

Standardized communications between modules

Three kinds of **exchange**:

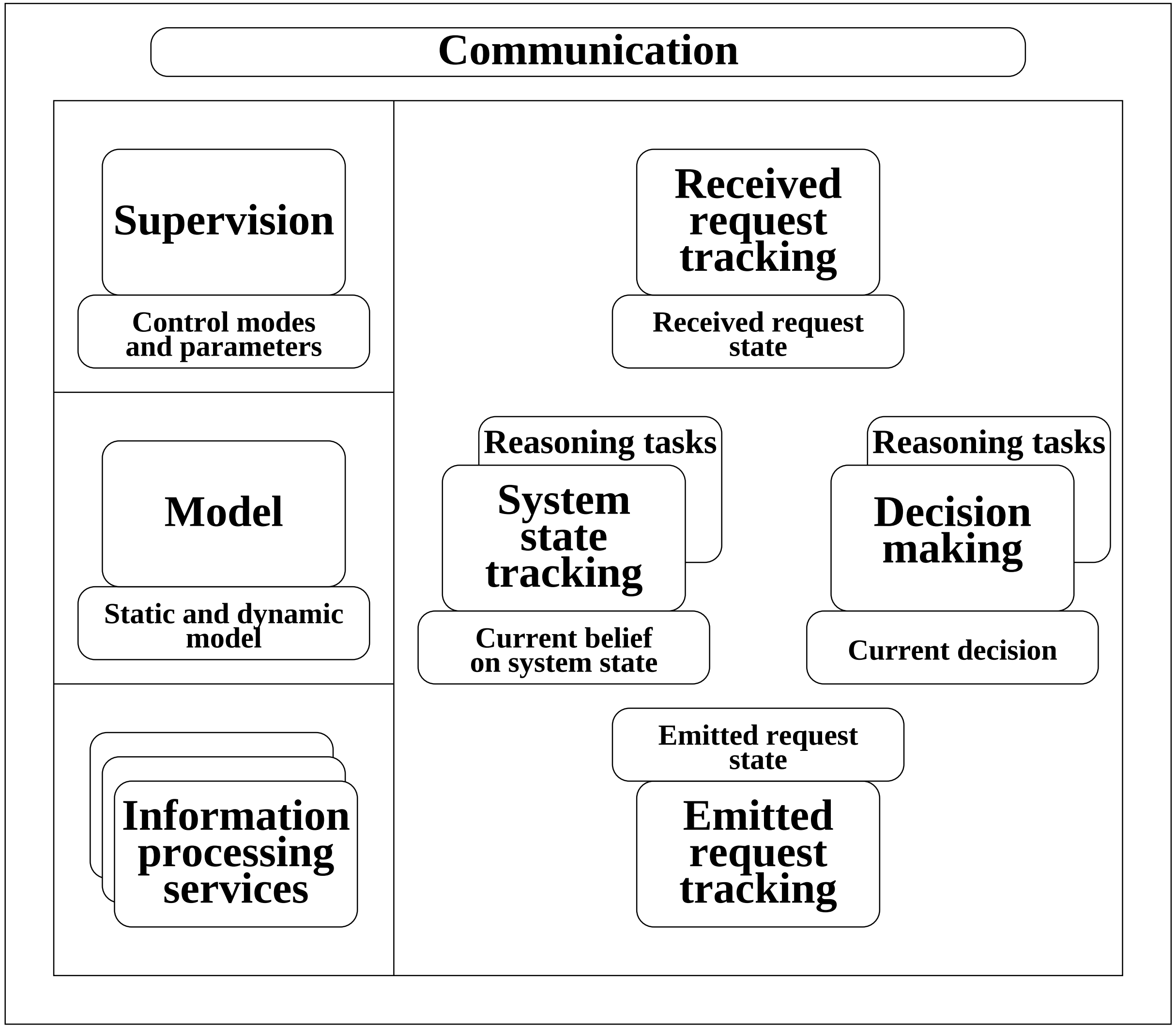
1. $\downarrow$ **control requests**
   emitted from a module to a lower level one;

2. $\uparrow$ **request reports**
   emitted in the opposite direction from a module to a higher level one;

3. $\uparrow$ **information** about the **system state**
   from a module to a higher level one.

Generic organization of each module

Control module



Generic scheme of interaction between reactive and deliberative tasks

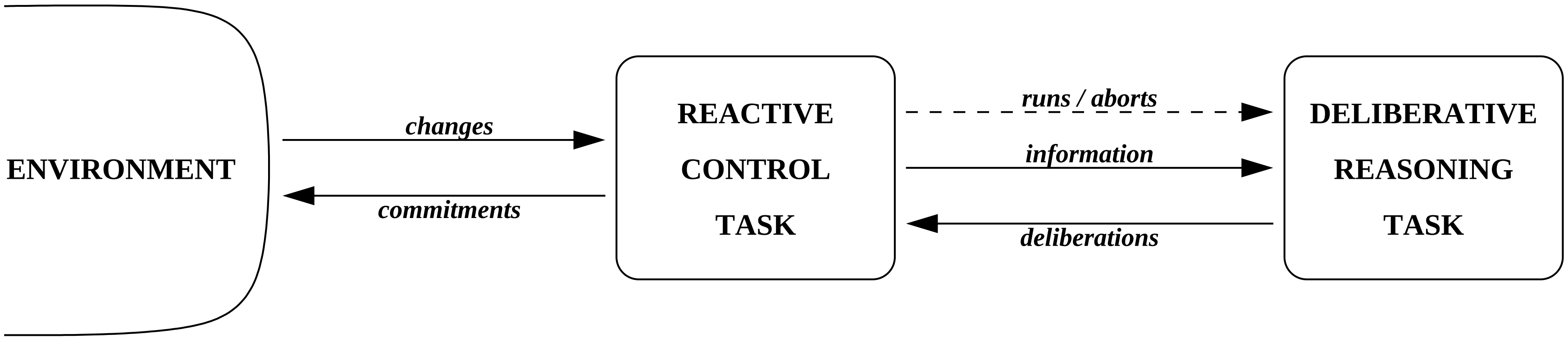
Need for a globally **reactive** control.

In some modules, **system state tracking** and **decision-making** may require calling **deliberative** tasks.

Questions:

1. what must be their **temporal behavior**?
2. how must they **interact** with **reactive** tasks?

Generic scheme of interaction



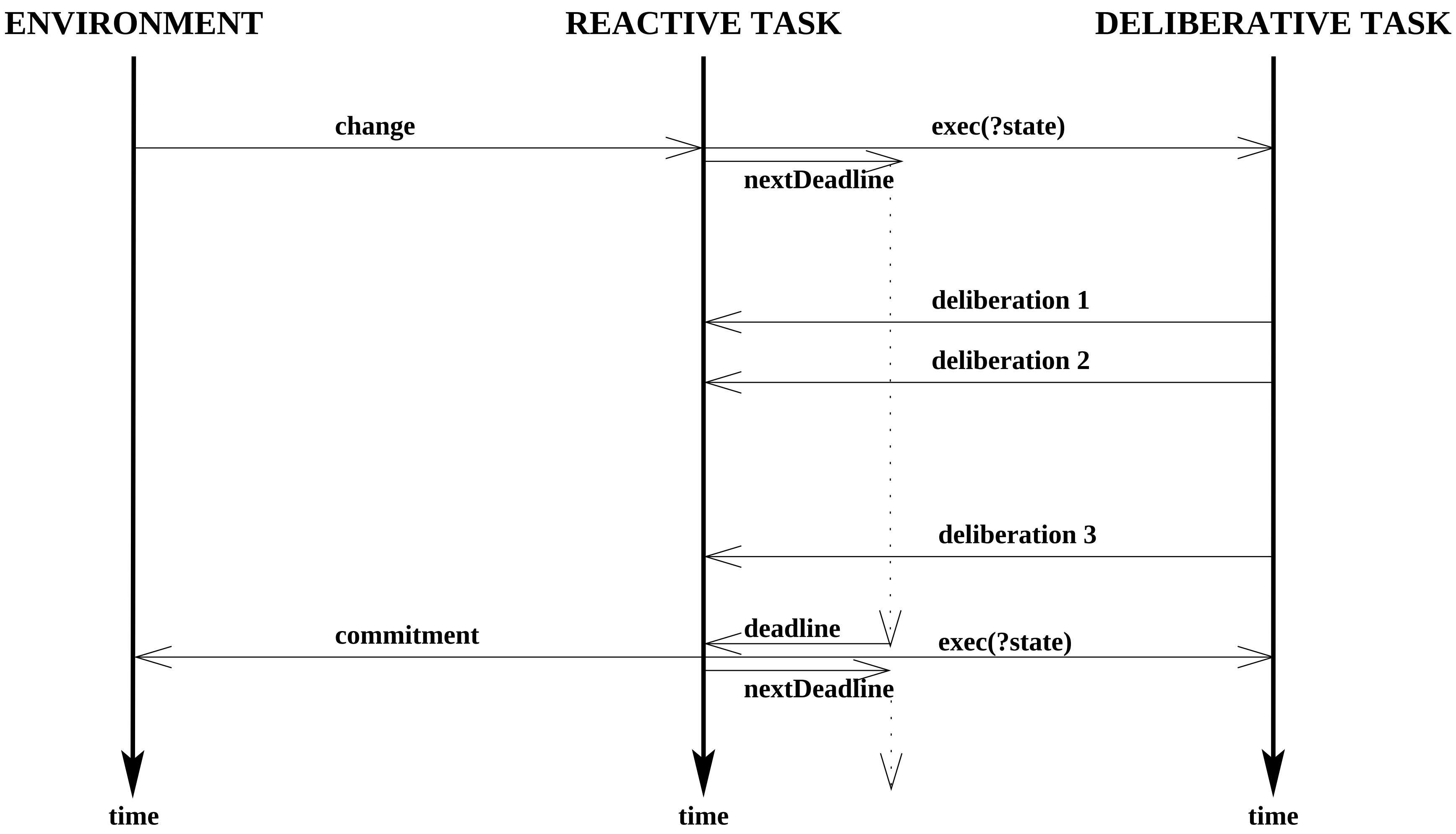
Assumptions about **deliberative** tasks:

1. **anytime** behavior.

Assumptions about **reactive** tasks. Ability:

1. to compute **deadlines**;
2. to **check deliberations** before decision-making;
3. to compute **default decisions** when no decision is available.

Example of scenario



Current implementation

Esterel for the **reactive** tasks.

Java for the **deliberative** ones.

Conclusion

Very useful architectural principles.

Applicable beyond the space domain.

Questions ?