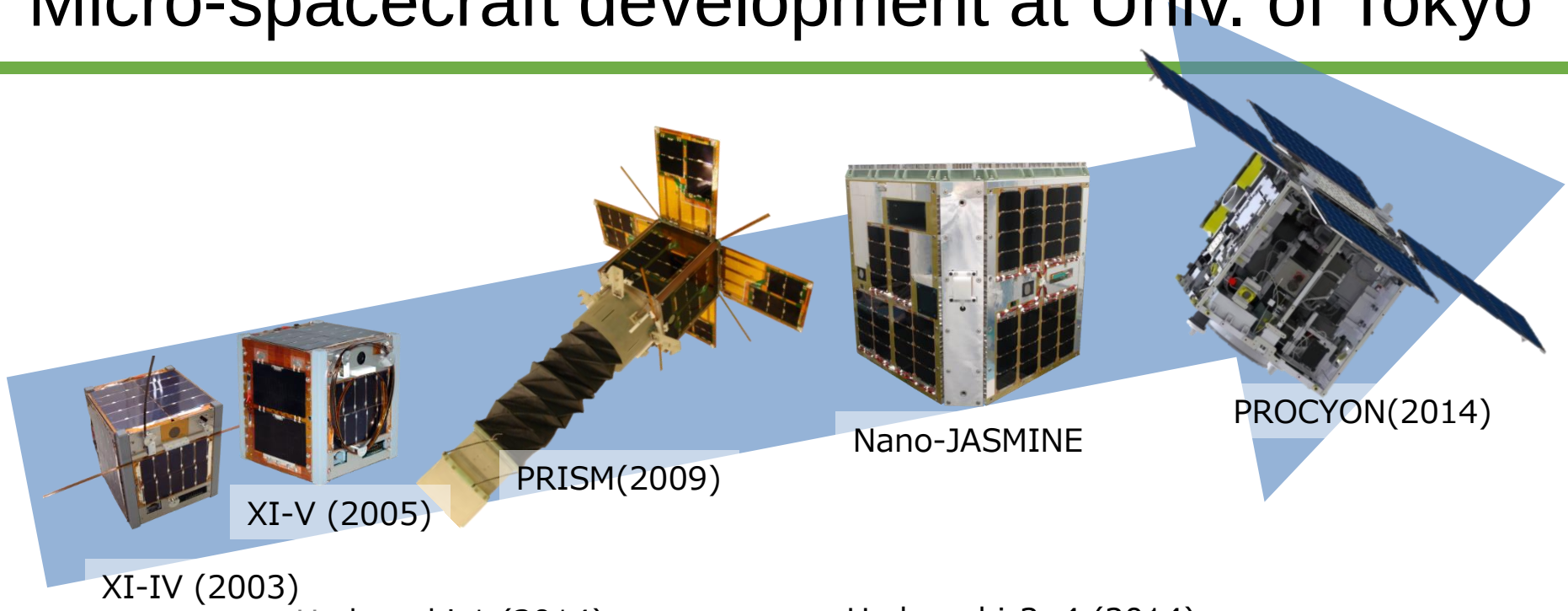
On-orbit Demonstration of Satellite Software Architecture with a Flexible Reconfiguration Capability

○**Shintaro Nakajima**, Satoshi Ikari, Masashi Tomooka,
Yoshihide Aoyanagi, Ryu Funase and Shinichi Nakasuka
(The University of Tokyo)

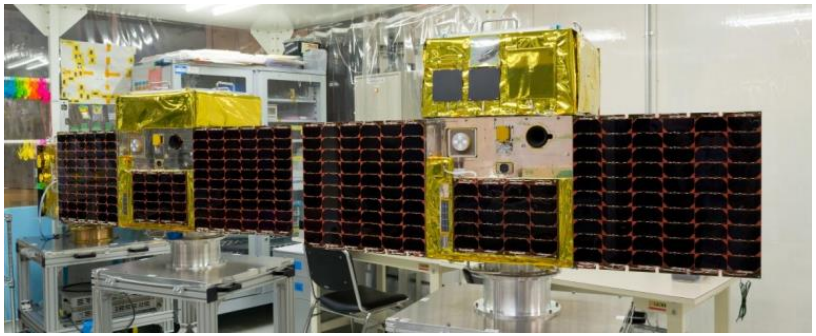
Introduction

Micro-spacecraft development at Univ. of Tokyo



Hodoyoshi-1 (2014)

Hodoyoshi-3, 4 (2014)



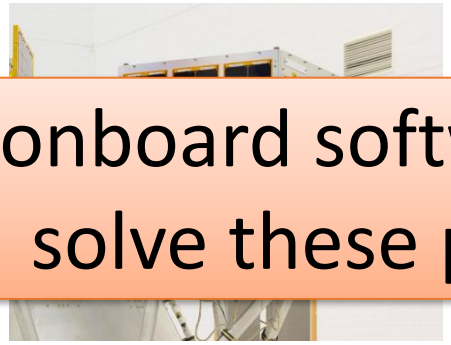
Micro-spacecraft development at Univ. of Tokyo

Since each onboard software(OBS) for each mission is developed independently, OBS

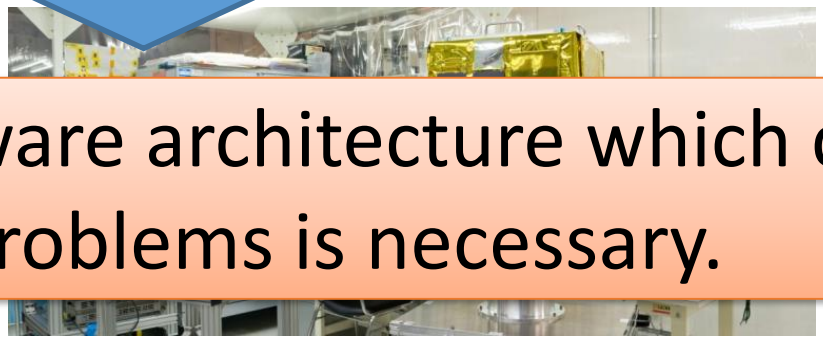
- needs long development term
- is less reliable

XI-IV (2003)

Hodoyoshi-1 (2014)



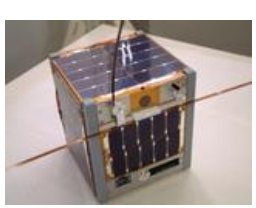
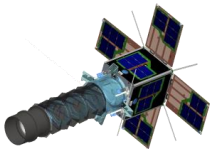
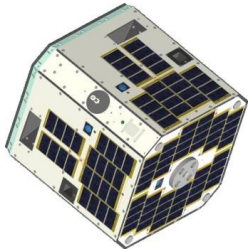
Hodoyoshi-3, 4 (2014)



New onboard software architecture which can solve these problems is necessary.

Introduction of new OBS architecture

- New OBS architecture for small spacecraft should have both reusability and on-orbit reconfigurability to achieve reliability and to shorten development term.

On-orbit reconfigurability

3. Development of new OBS architecture

Necessary function of OBS

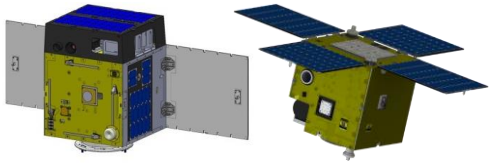
2. Integration

C2A

5. Applying to future spacecraft

4. On-orbit demonstration and evaluation

1. Knowledge extraction from previous spacecraft



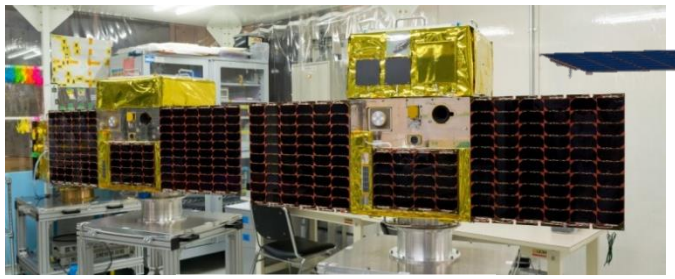
Derivation of OBS Architecture

Command Centric Architecture (C2A)[1]

[1] Doctoral dissertation by Takisawa, 2015

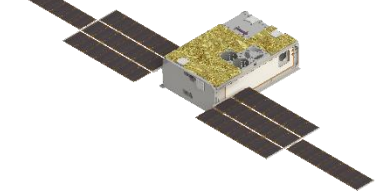
- Command Centric Architecture (C2A) is the software framework which realizes both reusability and flexible reconfigurability by describing ALL spacecraft's functions as "commands".
- C2A has been used for 3 micro-satellites and 1 interplanetary micro-spacecraft on-orbit, and 3 or more spacecraft being developed.

© UNIFORM, Wakayama Univ.



Hodoyoshi-3, 4

PROCYON

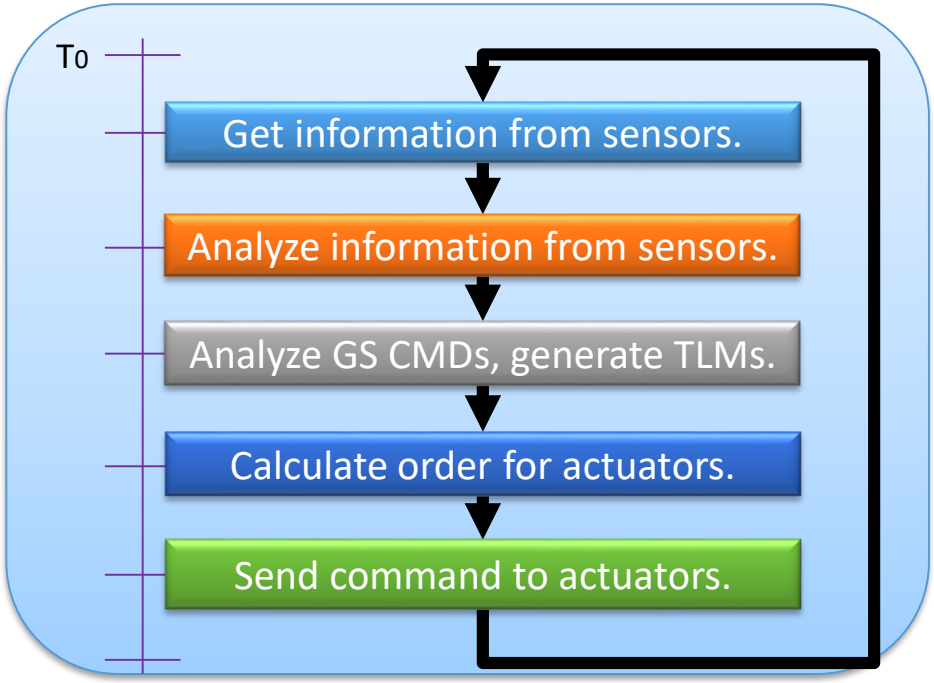


EQUULEUS

Common pattern of spacecraft's function #1

Common pattern of spacecraft's function :
Executing "pre-defined block of processing" at a specific timing.

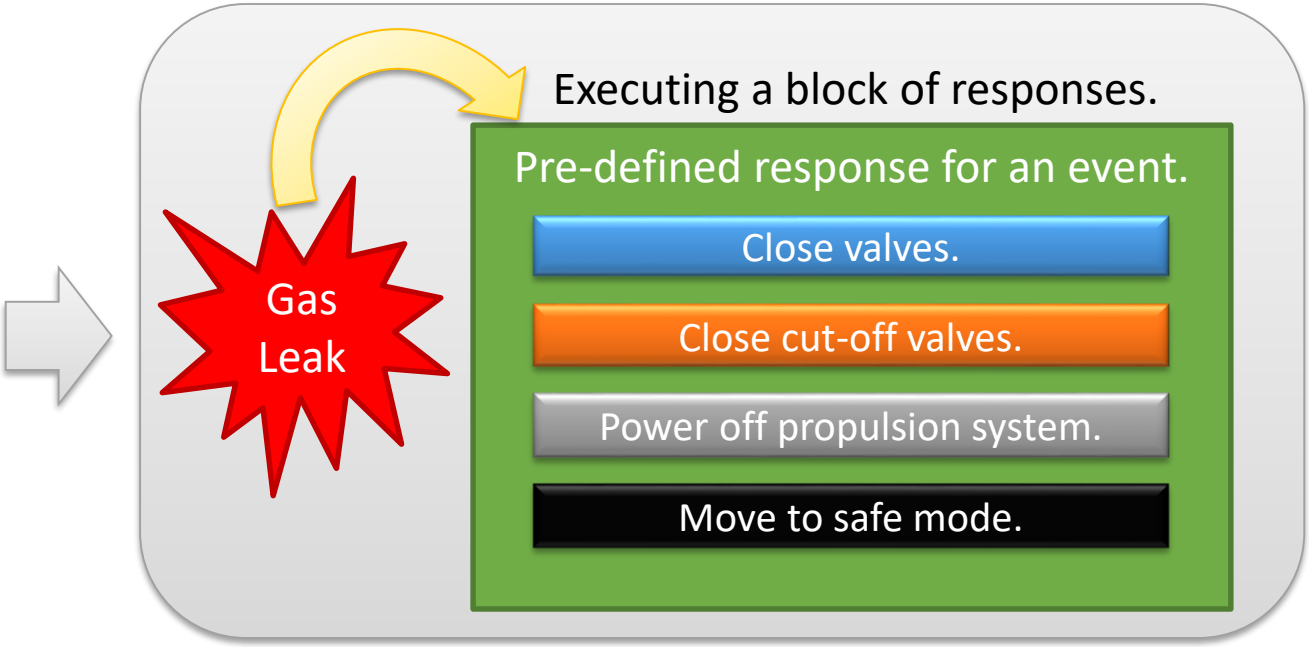
Ex.1 : Main loop of spacecraft.
-> Periodic processing for management and control of spacecraft's state.



Common pattern of spacecraft's function #2

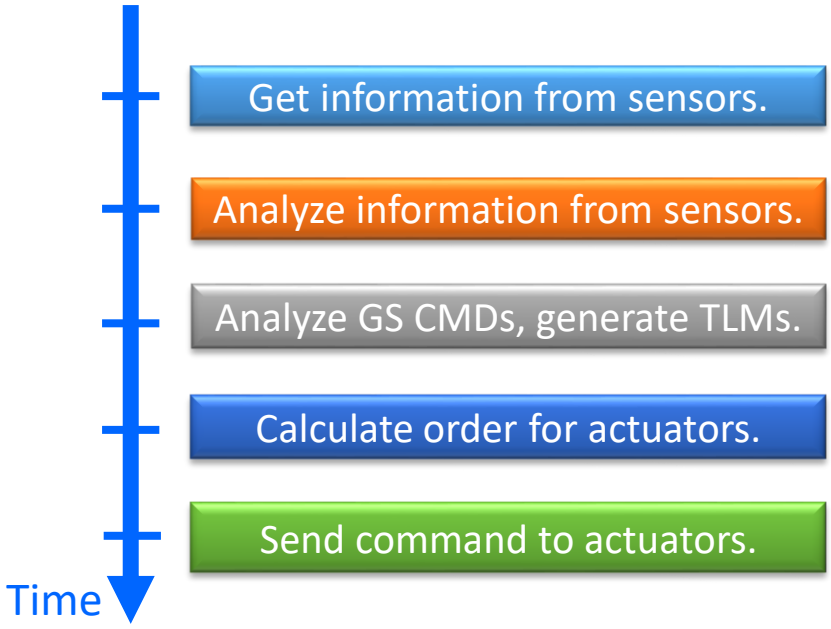
Common pattern of spacecraft's function :
Executing "pre-defined block of processing" at a specific timing.

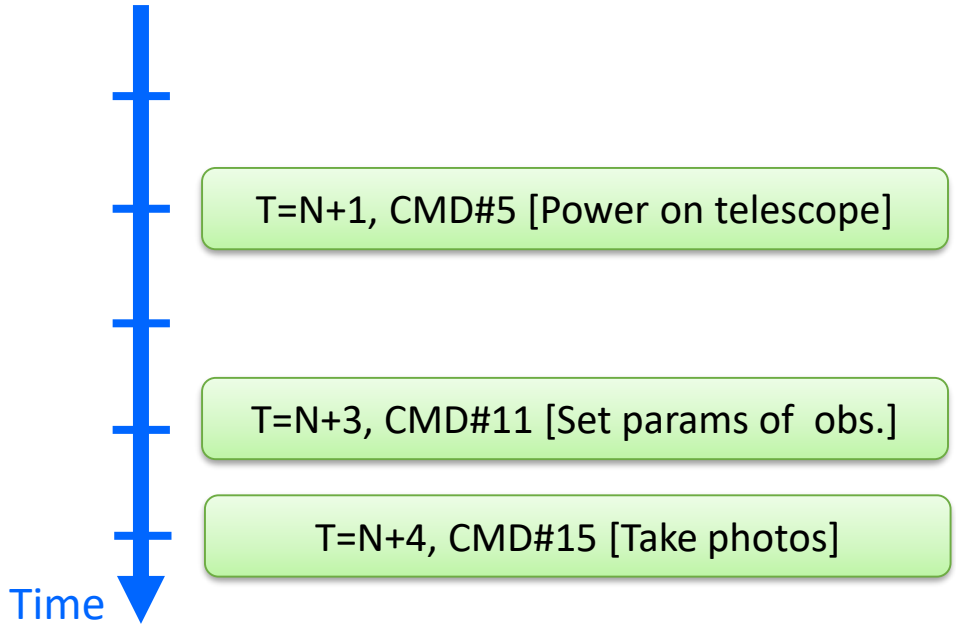
Ex.2: Response to accidental event.



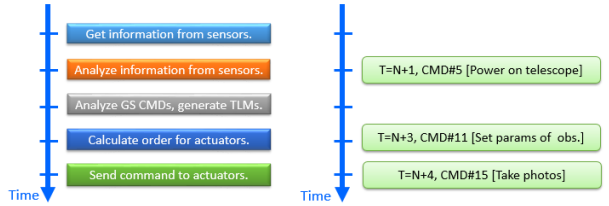
Idea of C2A

- Spacecraft's actions are performed specific time, and are defined as a group of specific functions.
- In other words, actions are very similar to time designated commands.





Idea of C2A

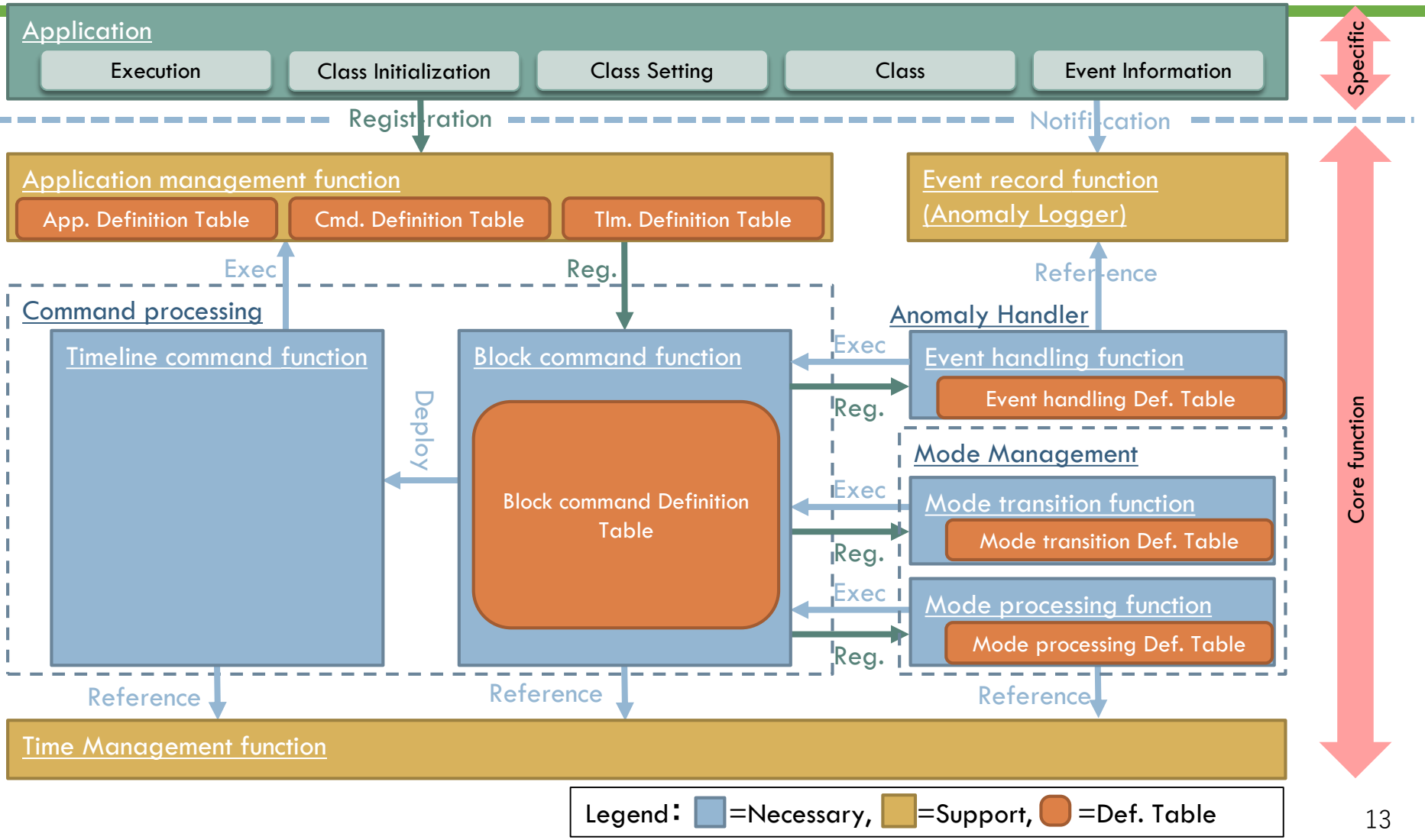


- Spacecraft's actions are performed specific time, and are defined as a group of specific functions.
- In other words, actions are very similar to time designated commands.
- Hence, software architecture which defines the behavior of satellite can be implemented based on "command" functions.
  - This "command" means not only an order from ground operator but also an order by itself.
- Also, this software architecture standardizes and modularizes each function.

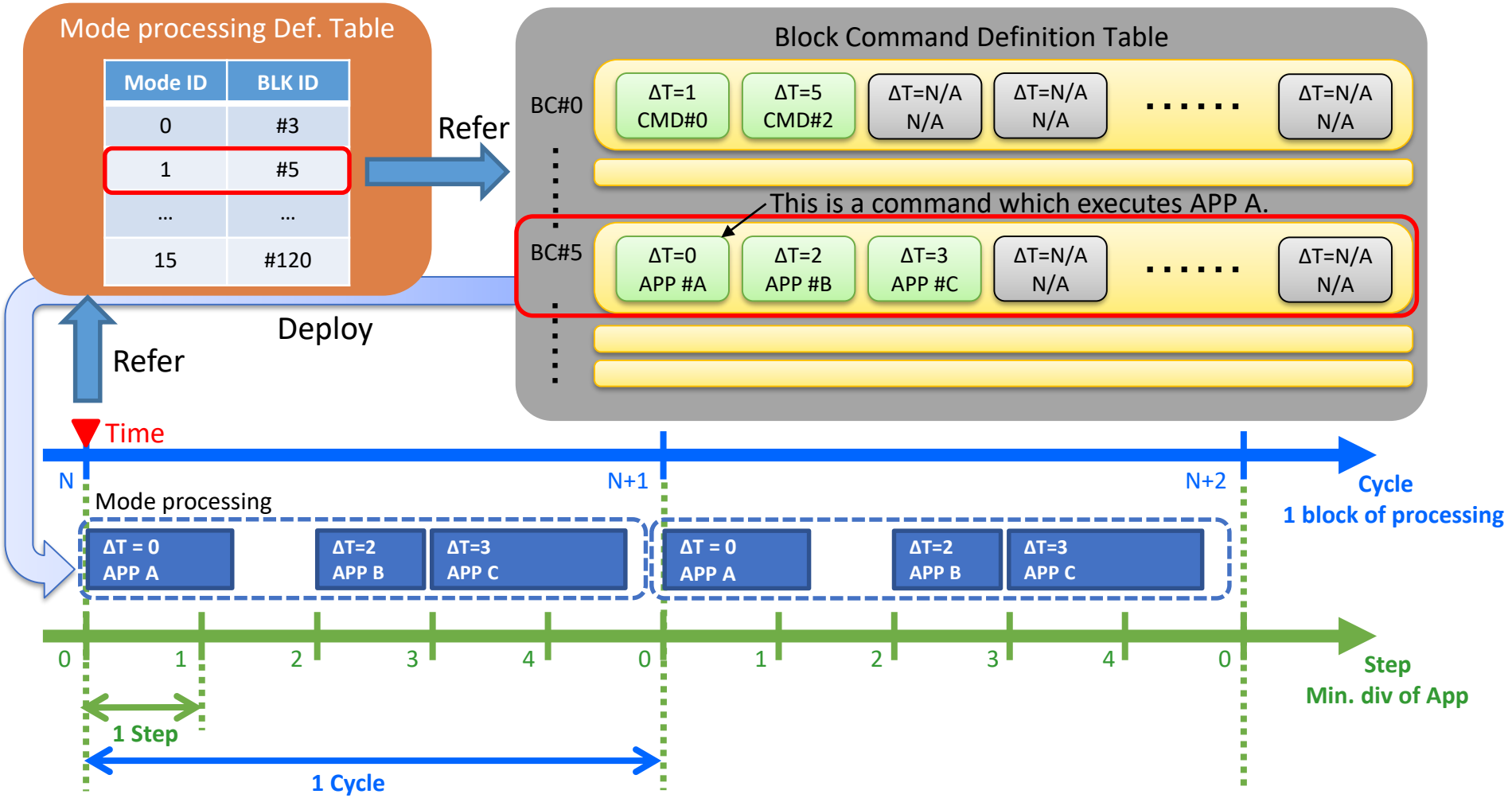
Specification of C2A

- Spacecraft's actions are defined as Block Commands (Macro Commands).
- Contents of Block Commands are described in re-writable "Definition Table."
- The relationships between functions are also described in the table.
- Common functions are called core functions, and satellite-specific functions are called applications.
  - Applications are standardized and modularized description method for functions in C2A.

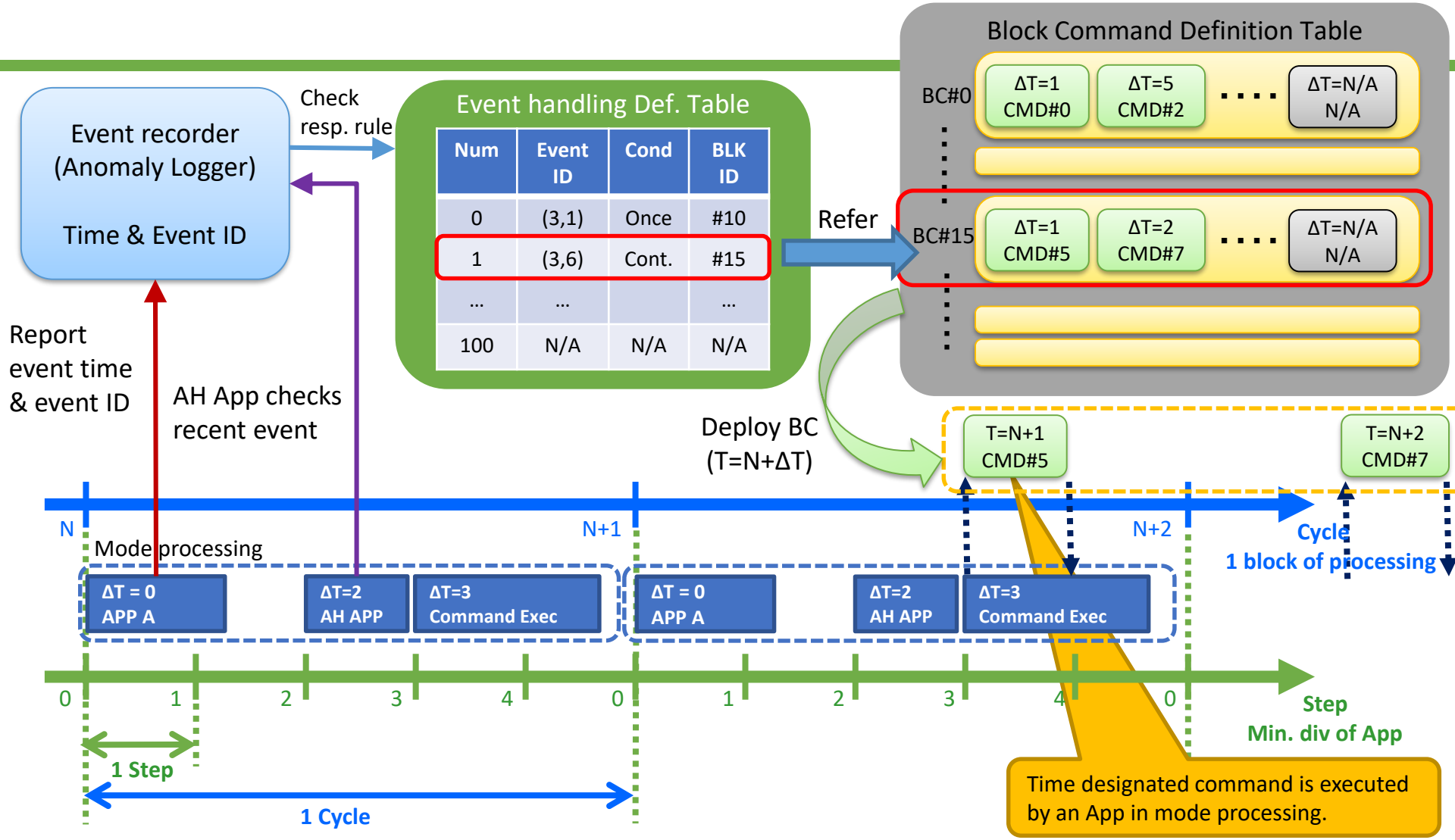
Schematic of C2A



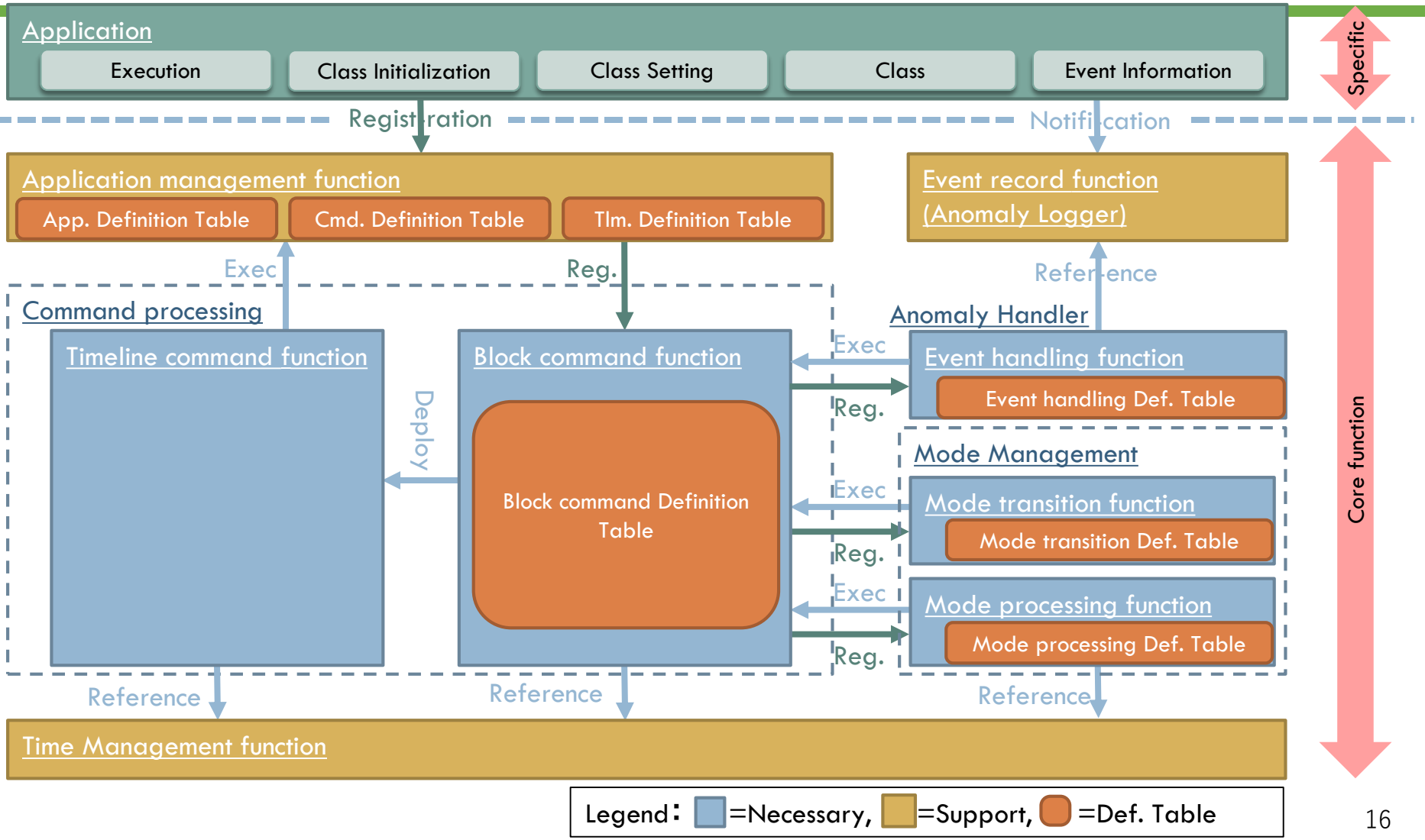
Example of C2A behavior: Main loop (Task List)



Example of C2A behavior: Anomaly Handler



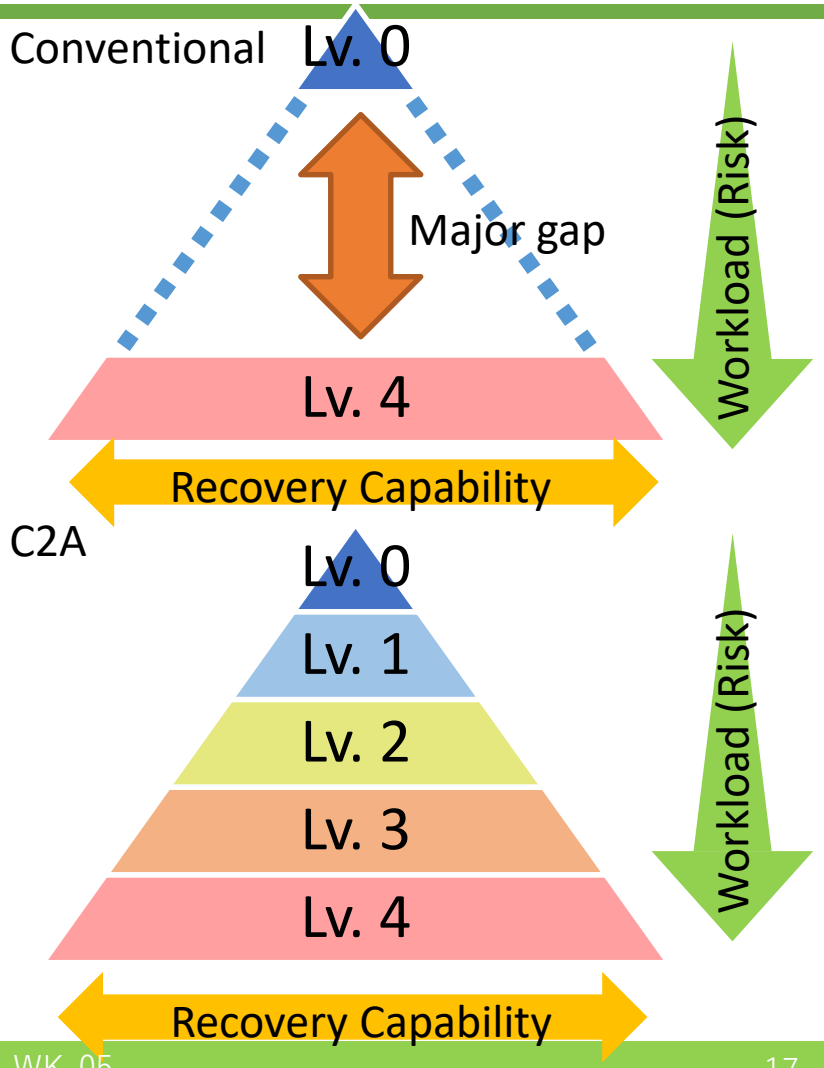
Schematic of C2A (Reshown)



Advantage of C2A
～Flexible Reconfigurability～

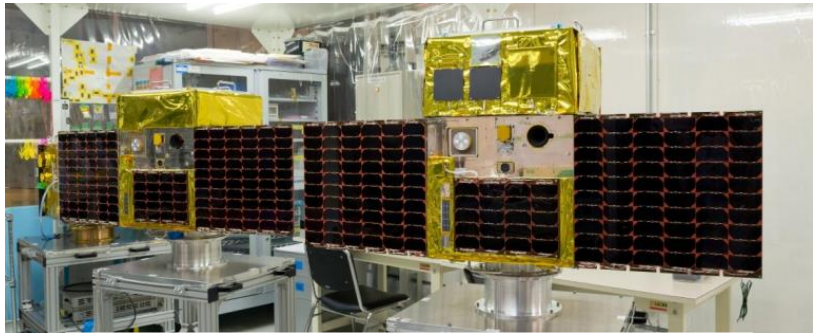
- Lv. 0 Changing parameters
- Lv. 1 Modifying block commands
  - Task List (main loop)
  - Mode transition
  - Event handling action
- Lv. 2 Modifying definition tables
  - Mode definition table
  - Mode transition table
  - Event handling table
- Lv. 3 Partially memory rewrite
  - Addition of new application.
- Lv. 4 Whole memory rewrite

C2A enables reconfiguration



On-orbit demonstration

Univ. of Tokyo Spacecraft using C2A



Hodoyoshi-3, 4 (60kg, June, 2014)

- Hodoyoshi-3, 4 are LEO micro satellites for Earth observation.
- C2A has been developed during development of these satellites.

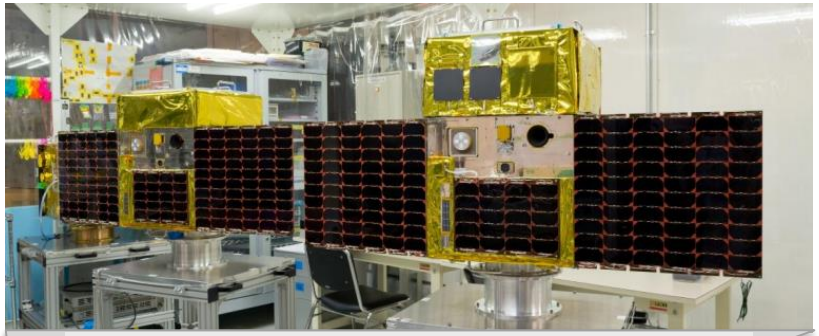
PROCYON(65kg, December, 2014)

- PROCYON is the interplanetary micro-spacecraft.
- C2A can be applied to the OBS of PROCYON from start of development.

Many bus components are common among these spacecraft.

We'd like to reuse some parts of OBS as much as possible by using C2A.

OBS development of PROCYON



Hodoyoshi-3, 4 (60kg, June, 2014)

PROCYON(65kg, December, 2014)

26th May
Beginning of OBS Dev.

16th July
S/W Integ. Begins

9th September
ADCS integration

6th November
Final OBS installation

Development schedule of PROCYON's OBS

End of Hodoyoshi's development

Individual funct. dev.

FM Environ. test▲

ADCS func. dev.

Integ. test▲

We have completed development of OBS in 5.5 months.

Comparison of Hodoyoshi with PROCYON

- Comparison in core functions

		Function	Command	Telemetry	Parameter
Auxiliary	Time management	Same	Same	Same	Modified
	Application management	Same	Same	Same	Modified
	Event logger	Same	Modified	Same	Modified
Necessary	Command dispatcher	Same	Same	Same	Modified
	Mode management	Same	Same	Same	Modified
	Event handler	Modified	Same	Same	Modified

Comparison of Hodoyoshi with PROCYON

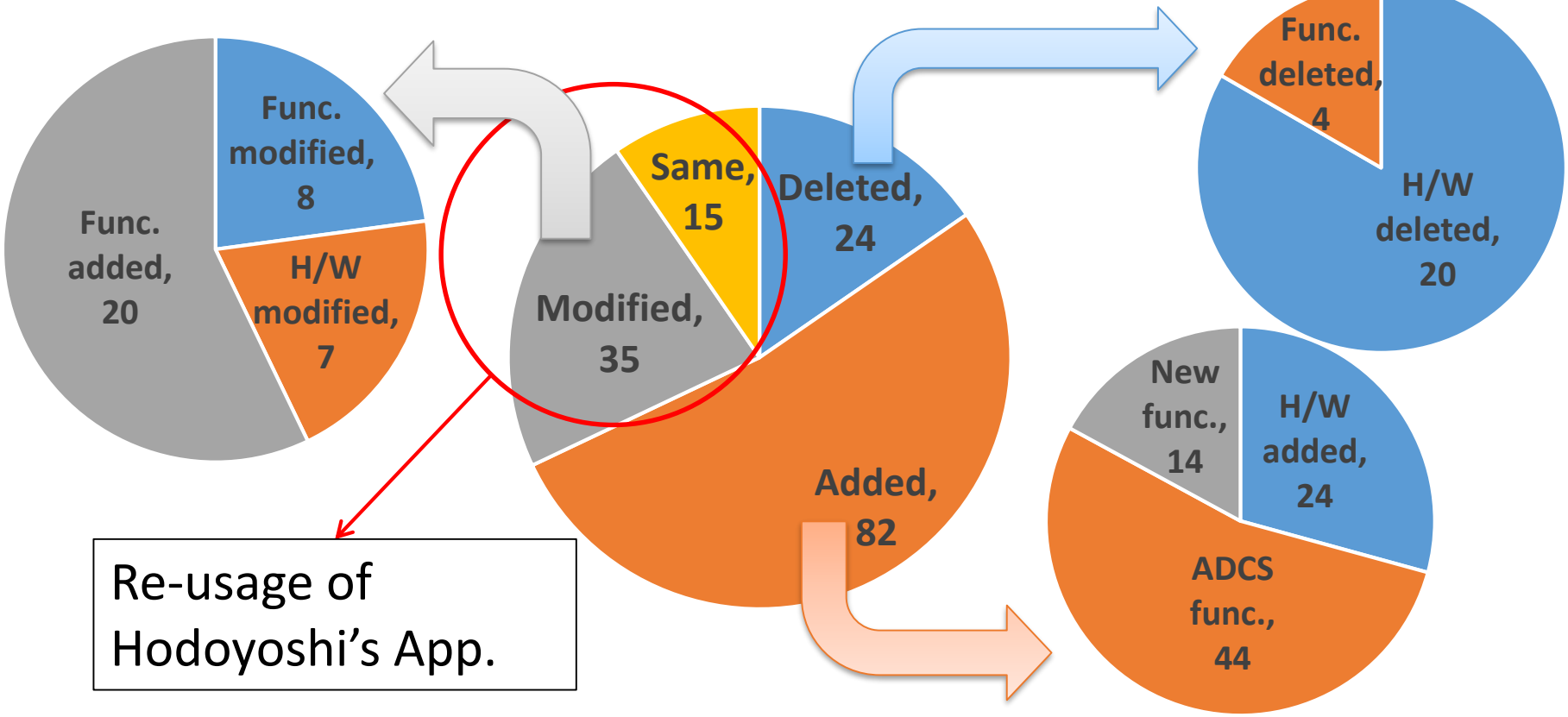
- Comparison in core functions

		Function	Command	Telemetry	Parameter
Auxi…				Same	Modified
Necessary	dispatche…				
	Mode management	Same			
	Event handler	Modified	Same	Same	Modified

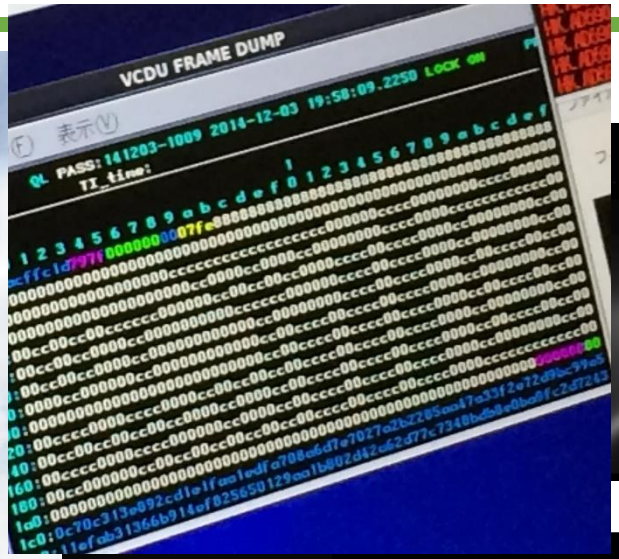
No functions are added or deleted.
Changes are due to the difference of number of OBC.
→Same software architecture can be applied to different spacecraft.

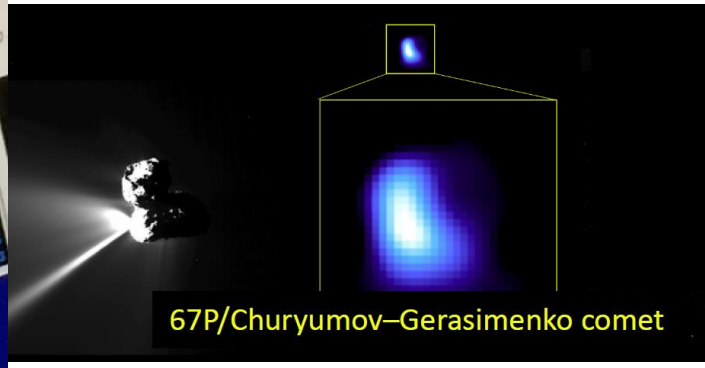
Comparison of Hodoyoshi with PROCYON

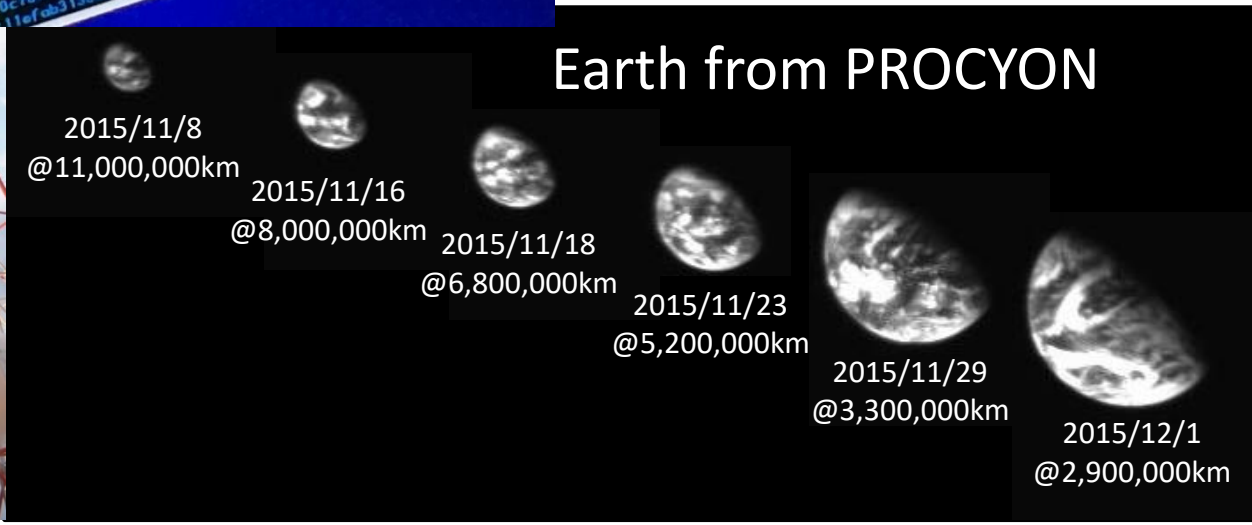
- Comparison of application: Many applications are added/deleted because of hardware modification.



Operation of PROCYON



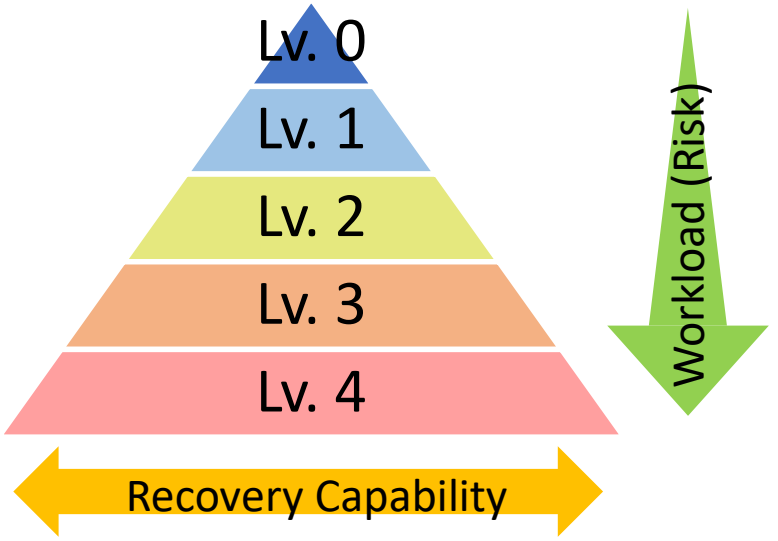




On-orbit reconfiguration of PROCYON OBS

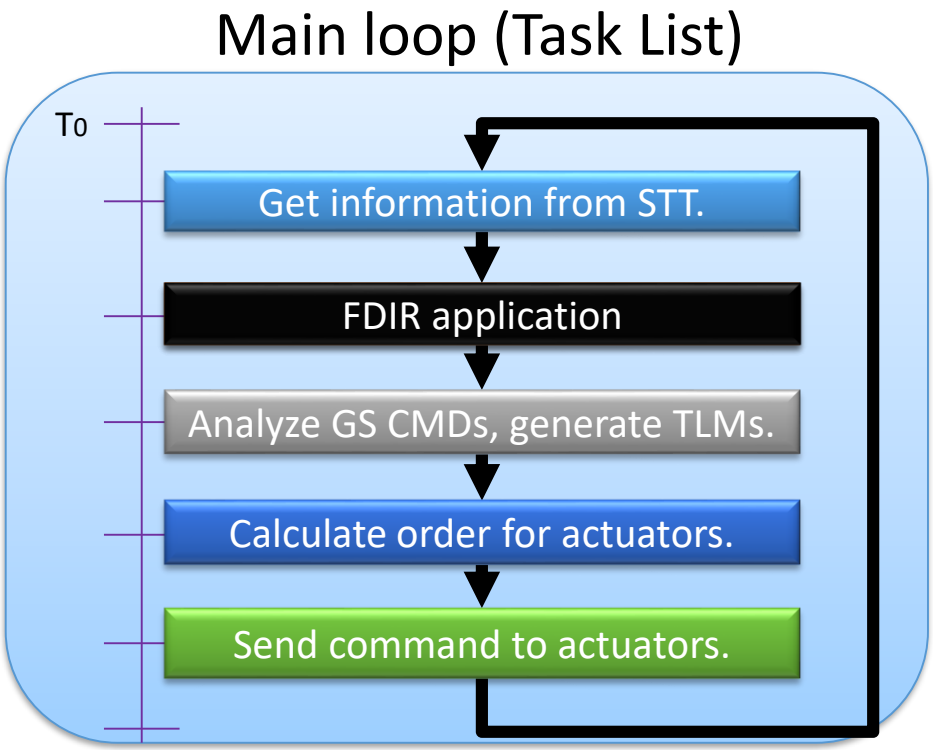
- We have performed many reconfigurations of PROCYON's onboard software.

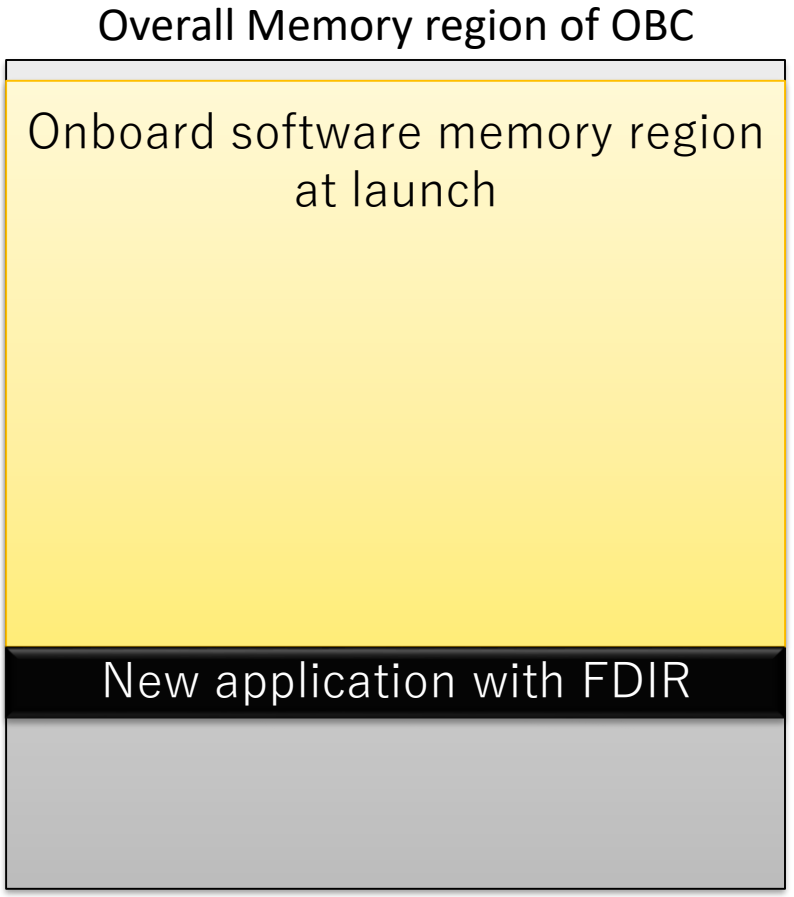
- Lv. 0: Changing parameters [Many times]
- Lv. 1: Modifying block commands [Many times]
  - Using combination of block commands for operation of propulsion system
- Lv. 2: Modifying definition tables [Some times]
  - Reconfiguration of TaskList (main loop) for observation by the telescope
- Lv. 3: Partially memory rewrite [4 times]
  - Adding FDIR function of STT
  - Adding application for angular velocity estimation using STT
  - Adding general purpose functions for anomaly detection and handling
  - Reconfiguration of HK telemetry packet
- Lv. 4: Whole memory rewrite [None]



Example of on-orbit reconfiguration

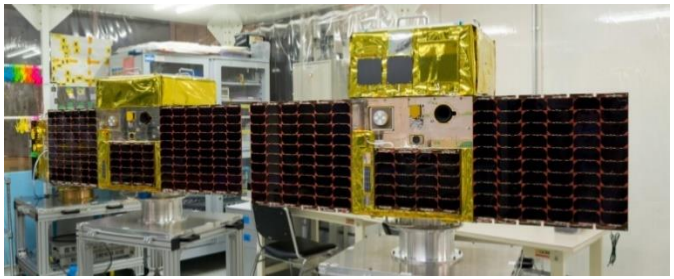
- Adding FDIR function of STT

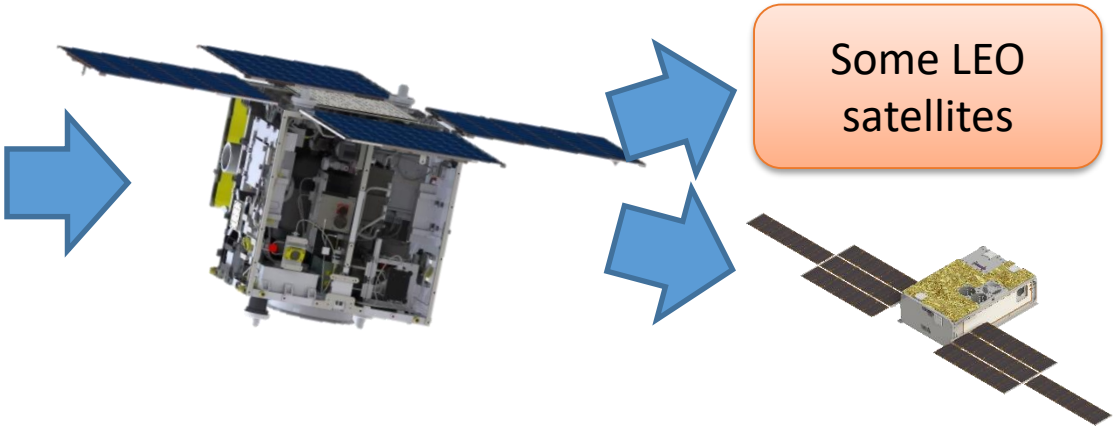




Conclusion

Future spacecraft using C2A





- After development of PROCYON, C2A has been implemented on some LEO satellites and one deep space CubeSat: EQUULEUS.
  - These satellites have different OBC, therefore C2A can be implemented different platforms.

Summary

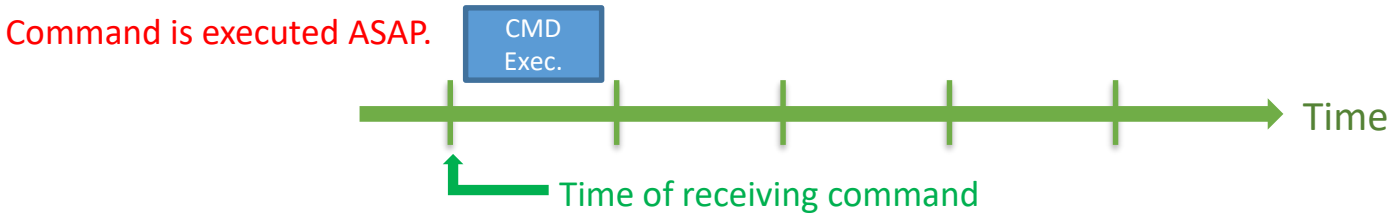
- We have developed Command Centric Architecture (C2A): on-board software architecture with flexible reconfigurability and reusability.
  - This architecture is based on command functions, especially block commands, and definition tables.
- C2A is used by some spacecraft on-orbit, Hodoyoshi satellites and PROCYON.
  - The r-usability of C2A was demonstrated during development of PROCYON, reconfigurability of C2A was demonstrated by operation of PROCYON.
- C2A will be used for future spacecraft in the Univ. of Tokyo and some other groups.

Contact: nakajima@space.t.u-tokyo.ac.jp / Booth:U8 (University of Tokyo)

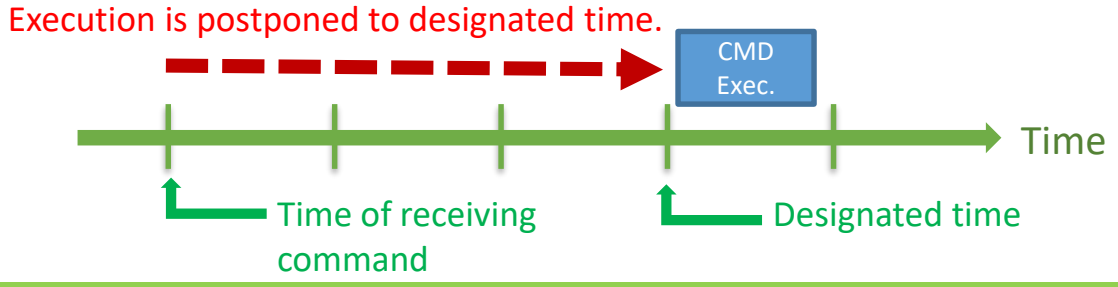
Back-up

Classification of command

- Real Time Command (RTC)
  - The command is executed at the time of receiving

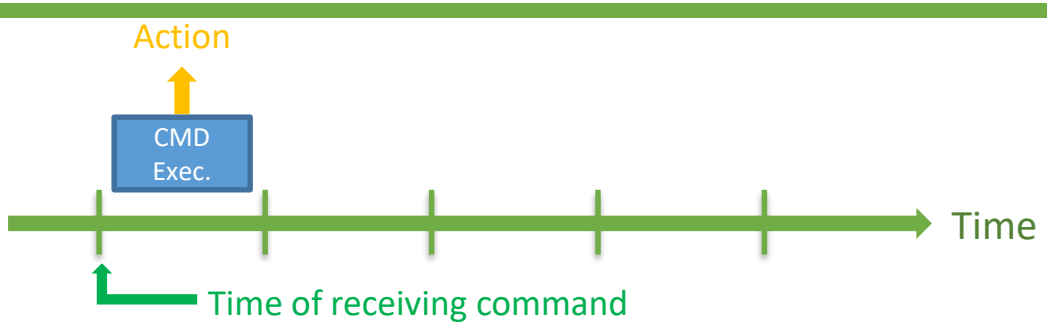


- TimeLine Command (TLC)
  - The command is executed at designated time.
  - This “time” is defined as time counter of OBC.
  - Spacecraft has a list of TLCs awaiting execution.

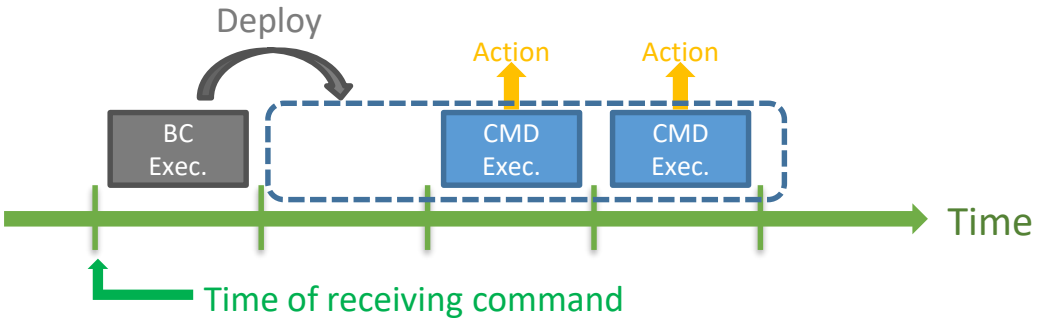


Classification of command

- Single Command
  - Executes one function



- Block Command (BC): Macro Command
  - BC is a group of some single commands (TLCs).
  - BC defines the order of command execution and reference time of execution based on the time of deploy.
  - To define BC is called “register,” and to execute BC is called “deploy.”
  - BC becomes some TLCs after it is deployed.



Command dispatcher

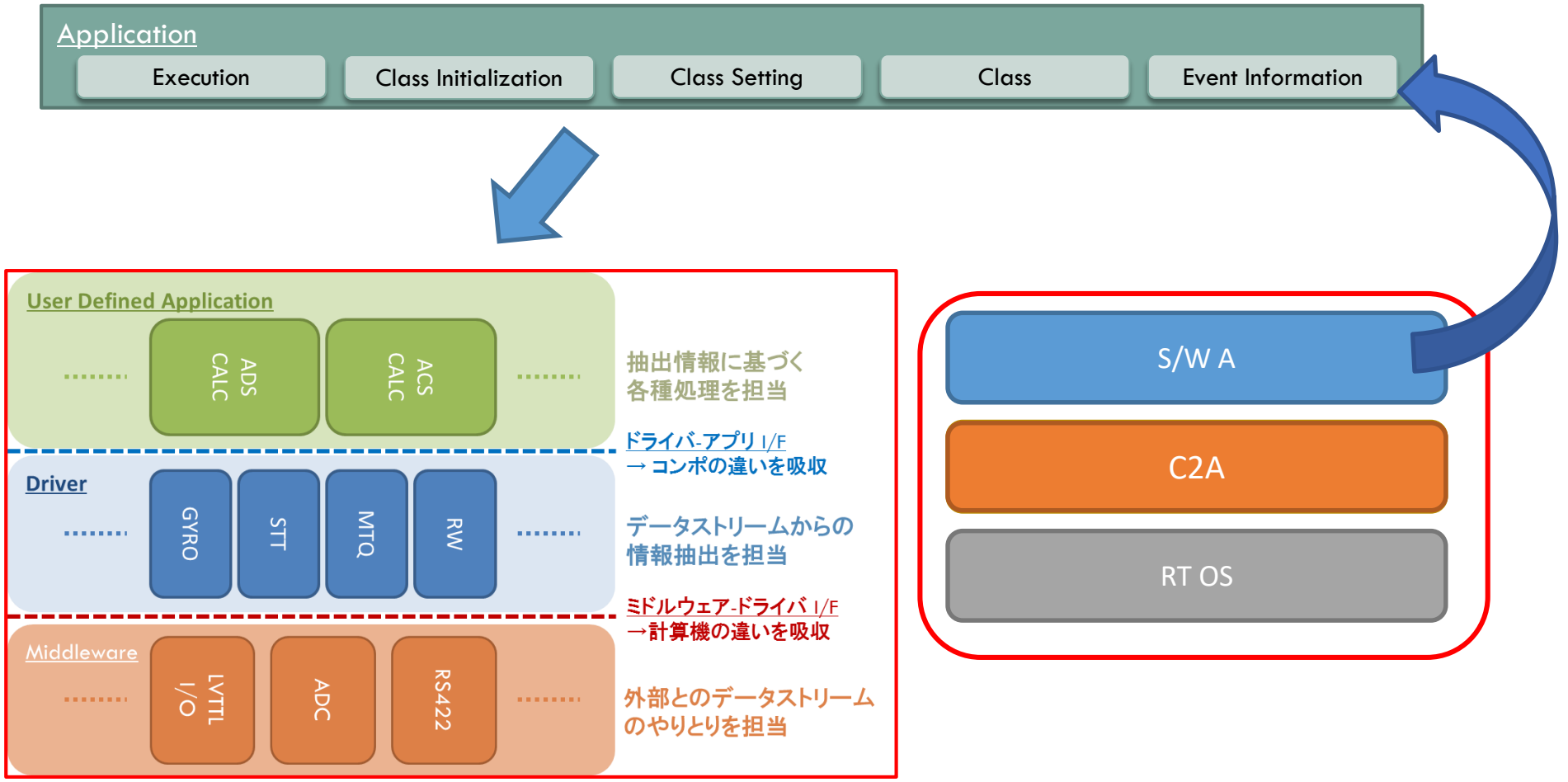
- Command means generally the order from the ground station to the satellite.
  - As a case of C2A, command means also the order from satellite itself.
  - In other words, command in C2A is a function in the program.
- Commands are classified to four types shown as a table below.

	Real Time	TimeLine
Single	Single and Real time	Single and TimeLine
Block	Block and Real time	Block and TimeLine

Core functions of C2A

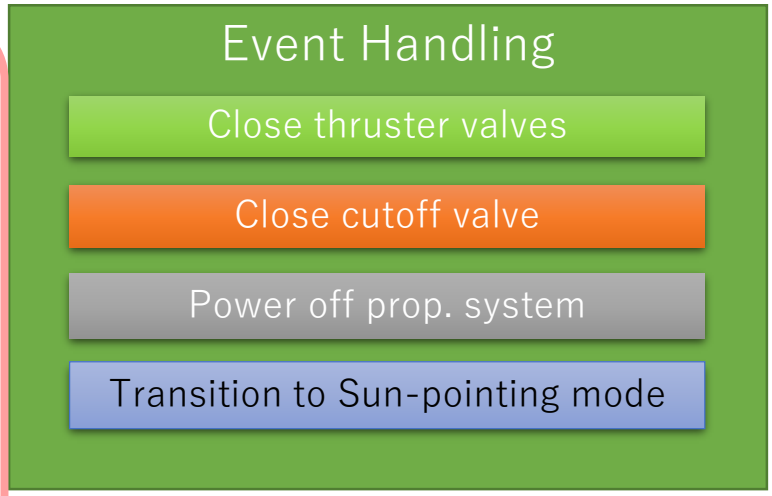
- Necessary functions: The functions which realizes common functions among any spacecraft.
  - Command dispatcher
  - Mode processing
  - Event handling
- Auxiliary functions: The functions to realize above "Necessary functions."
  - Time management
  - Application management
  - Event recorder

Layer structure of C2A



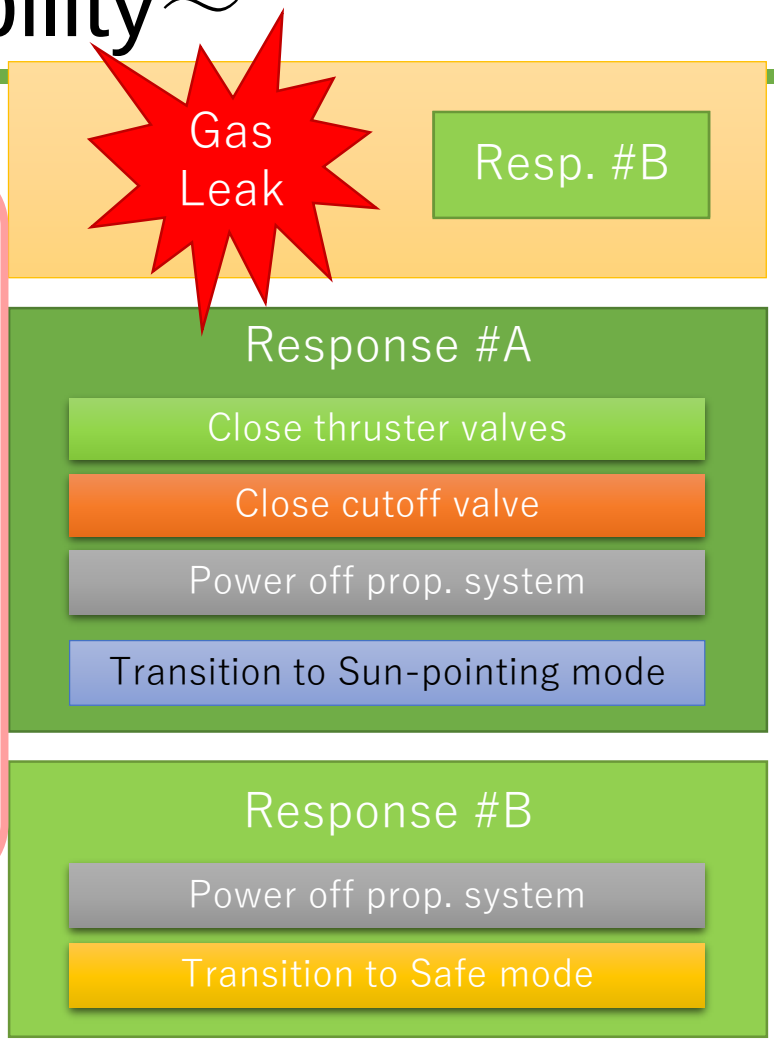
Advantage of C2A
～Flexible Reconfigurability～

- Lv. 0 Changing parameter
- Lv. 1 Modifying block command
  - Tasklist (main loop)
  - Mode transition
  - Event handling action
- Lv. 2 Modifying definition table
  - Mode definition table
  - Mode transition table
  - Event handling table
- Lv. 3 Partially memory rewrite
  - Addition of new application.
- Lv. 4 Whole memory rewrite



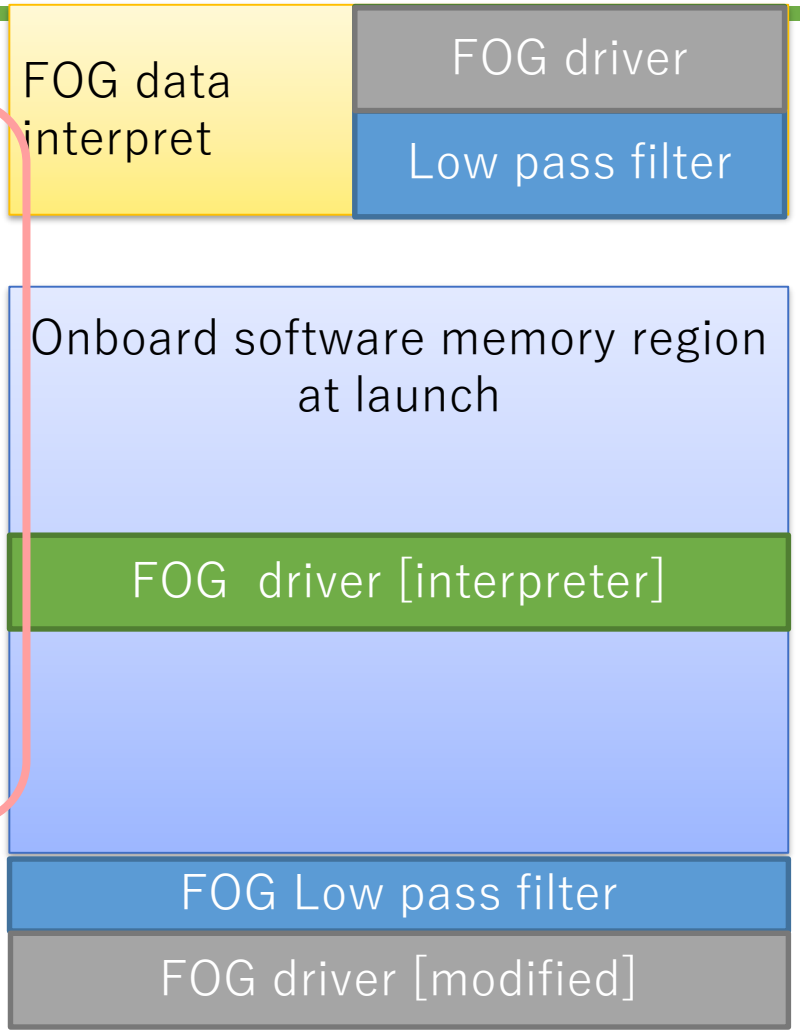
Advantage of C2A
～Flexible Reconfigurability～

- Lv. 0 Changing parameter
- Lv. 1 Modifying block command
  - Tasklist (main loop)
  - Mode transition
  - Event handling action
- Lv. 2 Modifying definition table
  - Mode definition table
  - Mode transition table
  - Event handling table
- Lv. 3 Partially memory rewrite
  - Addition of new application.
- Lv. 4 Whole memory rewrite



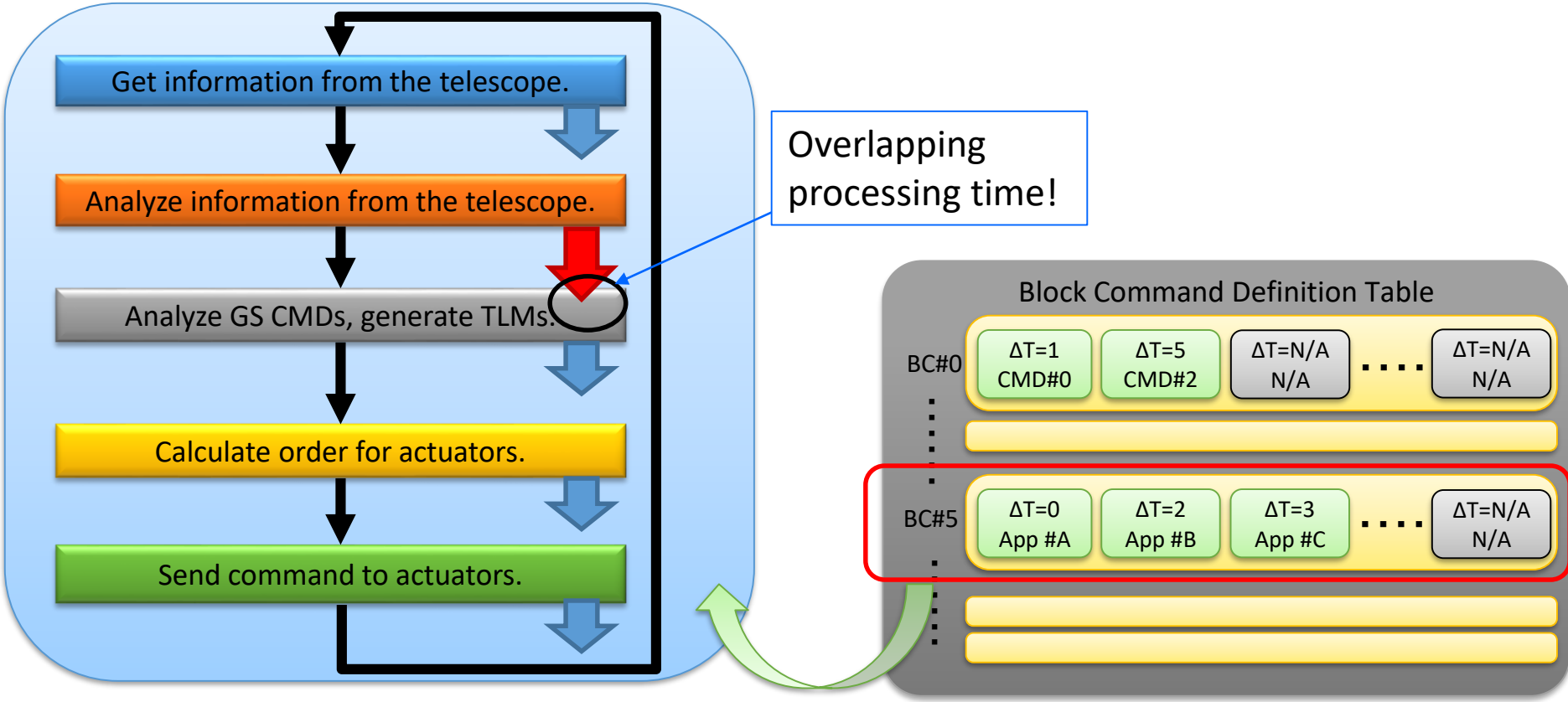
Advantage of C2A
～Flexible Reconfigurability～

- Lv. 0 Changing parameter
- Lv. 1 Modifying block command
  - Tasklist (main loop)
  - Mode transition
  - Event handling action
- Lv. 2 Modifying definition table
  - Mode definition table
  - Mode transition table
  - Event handling table
- Lv. 3 Partially memory rewrite
  - Addition of new application.
- Lv. 4 Whole memory rewrite



Example of on-orbit reconfiguration

- Reconfiguration of TaskList (main loop) for observation by the telescope



Example of on-orbit reconfiguration

- Reconfiguration of TaskList (main loop) for observation by the telescope