ERTS 2020

Formal Approach for the Verification of Onboard Autonomous function in Observation Satellites

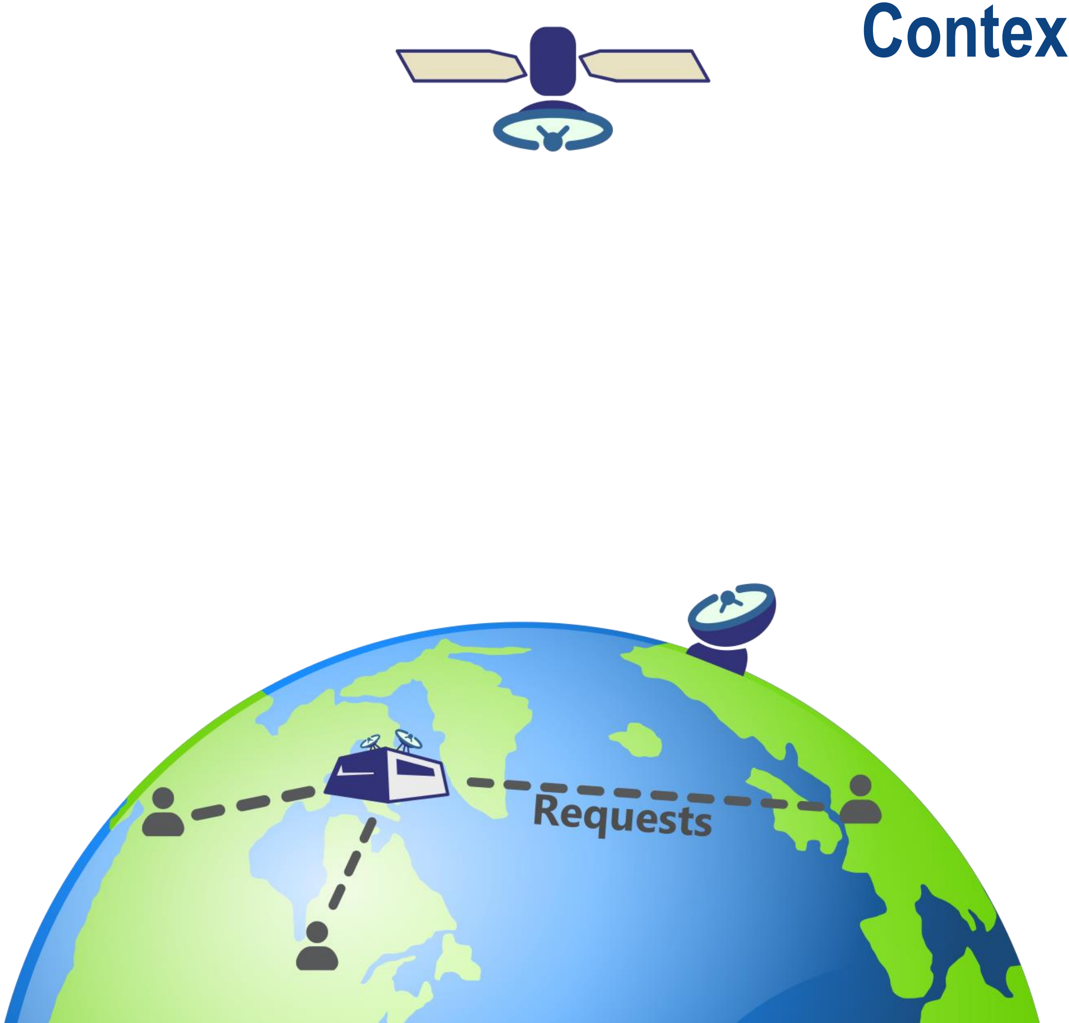
V. Mussot, S. Dal Zilio, L. Correnson, S. Rainjonneau, Y. Bardout, G. Scano

Summary

- Introduction
  - Context
  - Autonomy
- Formal Approach
  - Specification
  - Formal Model
- Experimentation
  - Framework
  - Prototype
- Conclusion

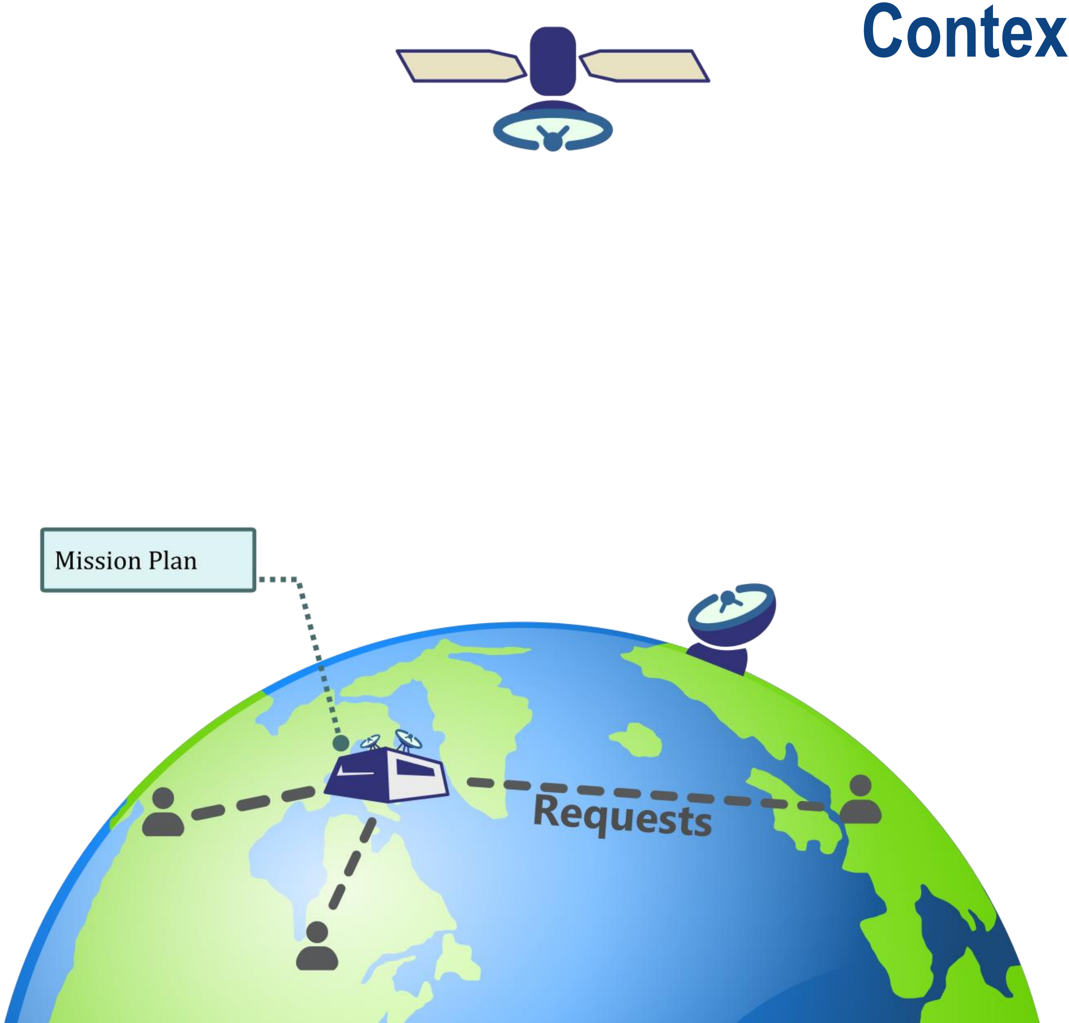
Context

- User requests gathering

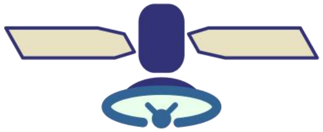


Context

- User requests gathering
- Mission Plan Building

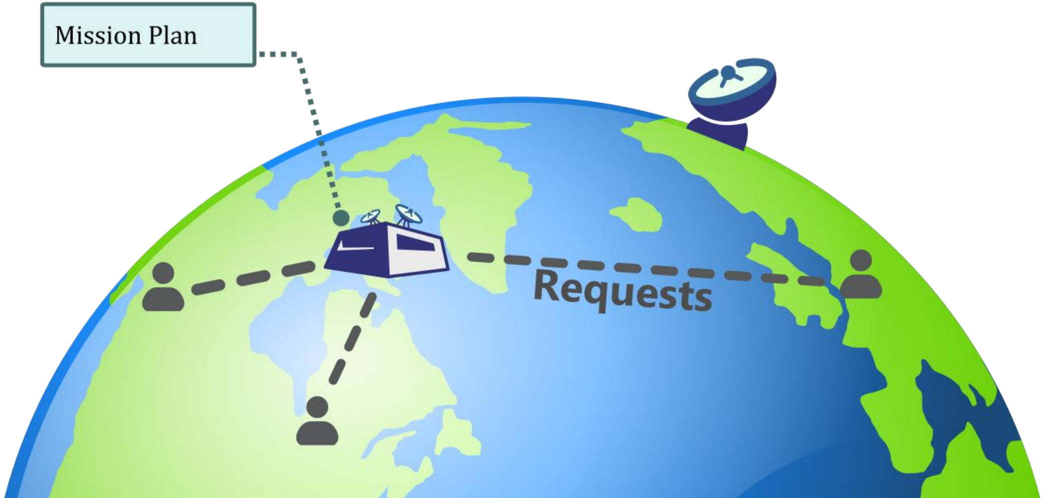


Context

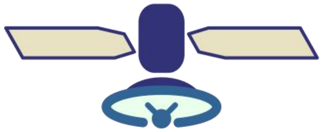


- User requests gathering
- Mission Plan Building
  - Sequence of time-tagged telecommands

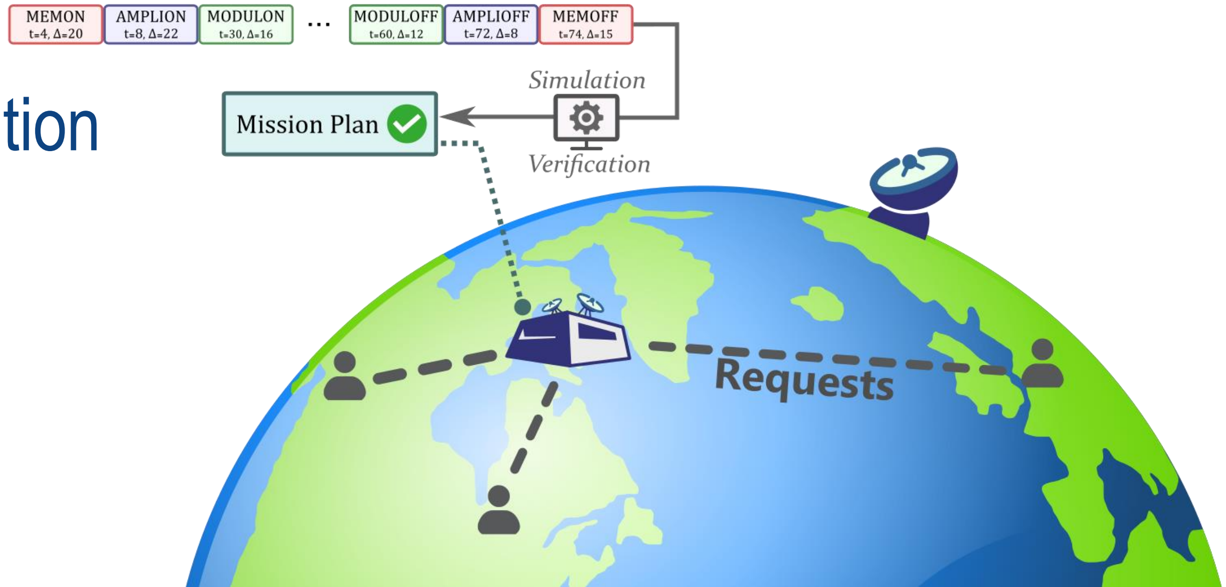
MEMON t=4, Δ=20	AMPLION t=8, Δ=22	MODULON t=30, Δ=16	…	MODULOFF t=60, Δ=12	AMPLIOFF t=72, Δ=8	MEMOFF t=74, Δ=15



Context



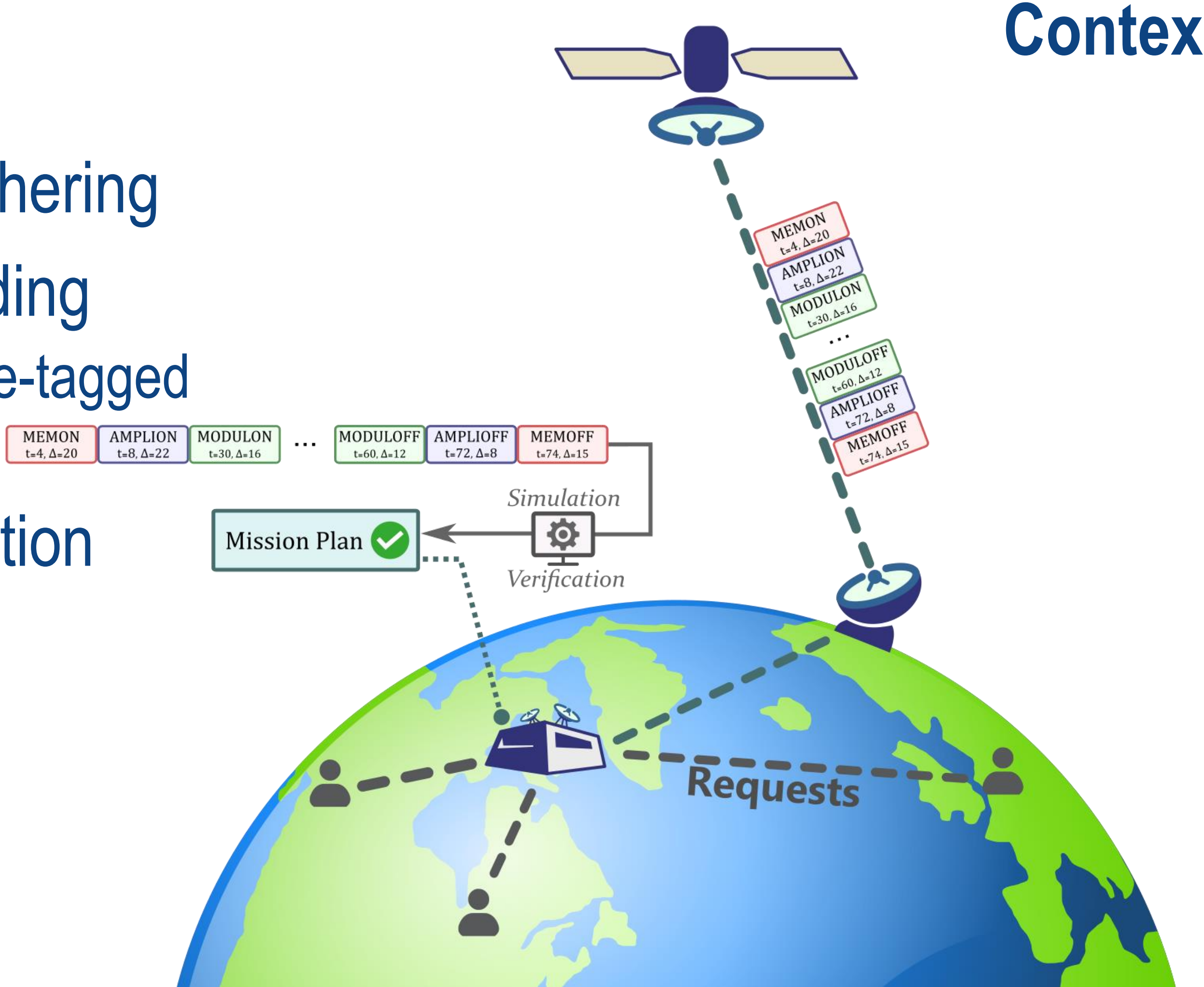
- User requests gathering
- Mission Plan Building
  - Sequence of time-tagged telecommands
- Sequence verification



MEMON t=4, Δ=20 | AMPLION t=8, Δ=22 | MODULON t=30, Δ=16 | … | MODULOFF t=60, Δ=12 | AMPLIOFF t=72, Δ=8 | MEMOFF t=74, Δ=15

Context

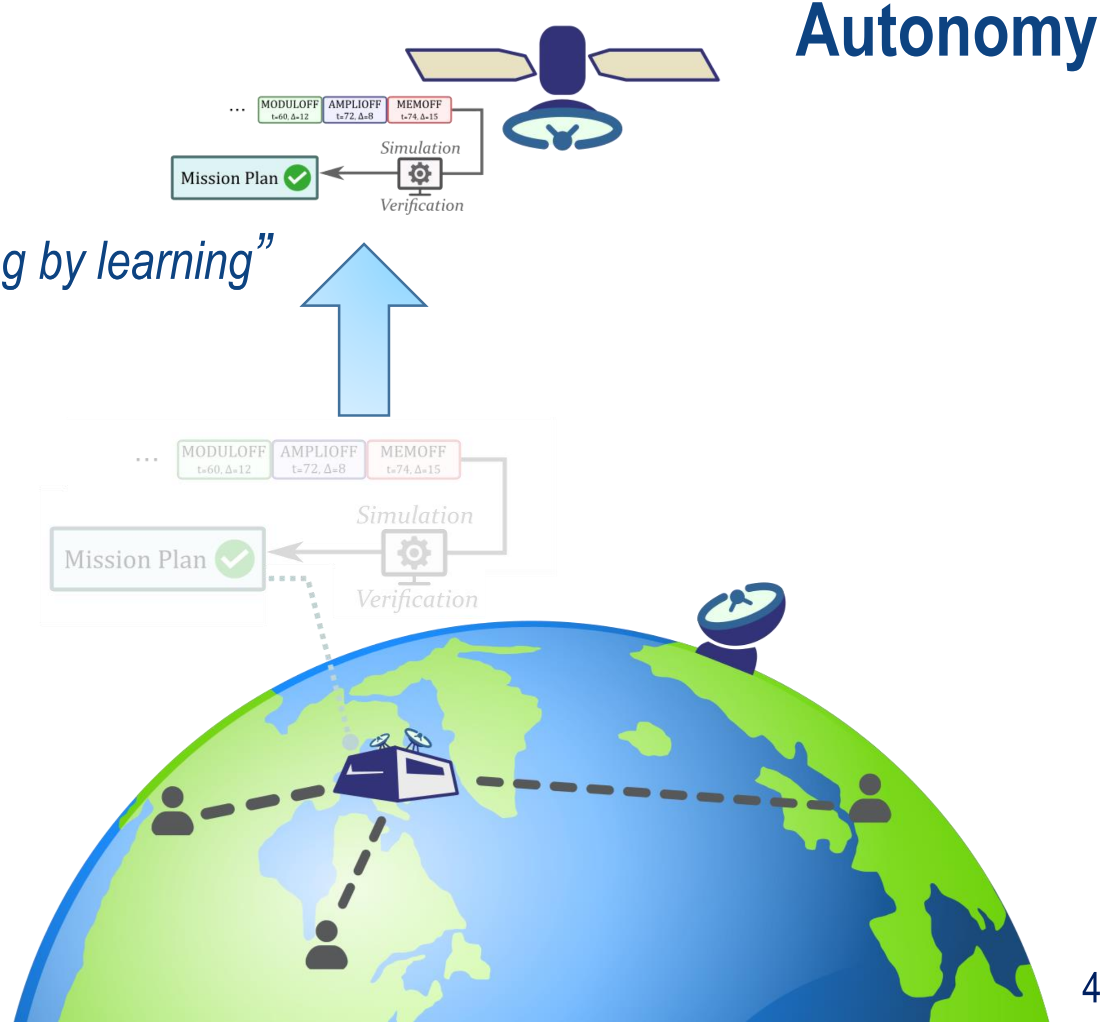
- User requests gathering
- Mission Plan Building
  - Sequence of time-tagged telecommands
- Sequence verification
- Plan upload



Autonomy

SYNAPSE:

“System autonomy & data processing by learning”



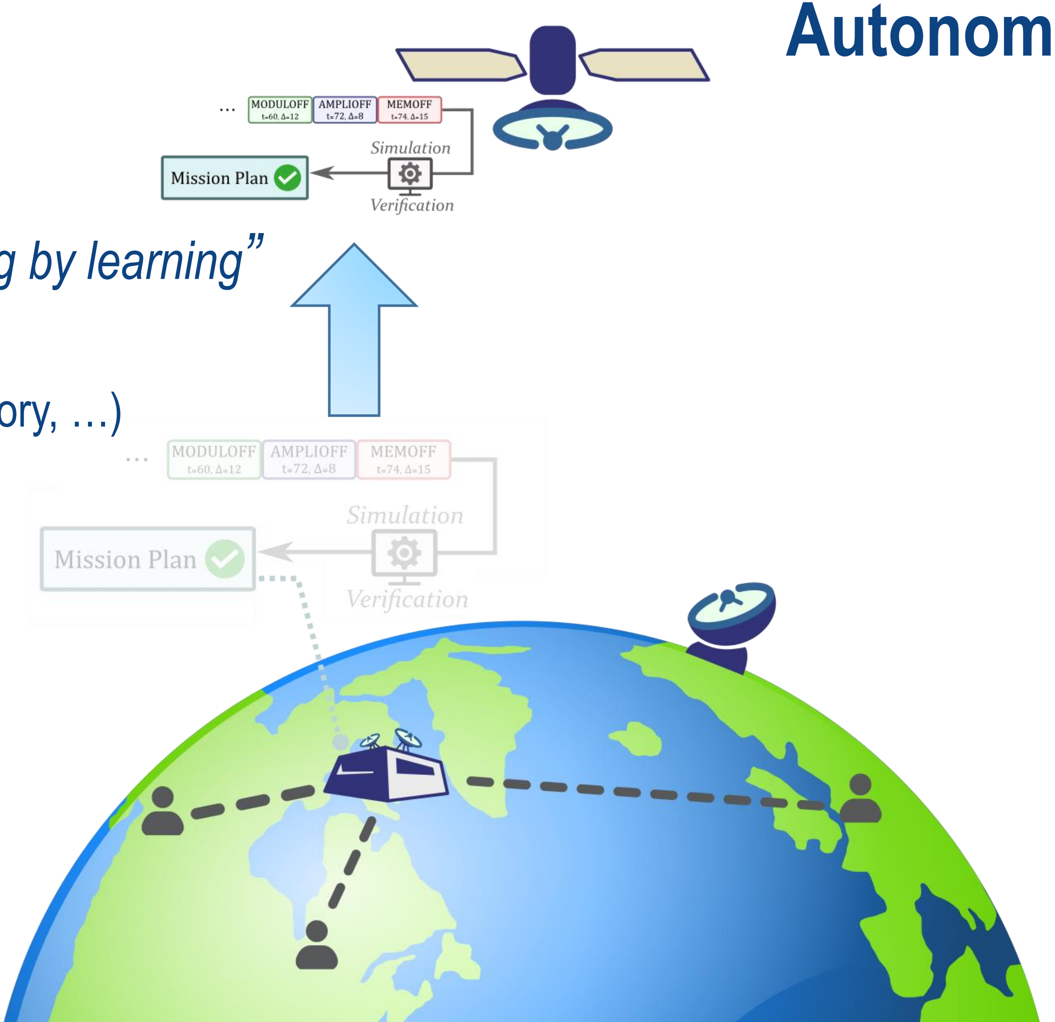
Autonomy

SYNAPSE:

“*System autonomy & data processing by learning*”

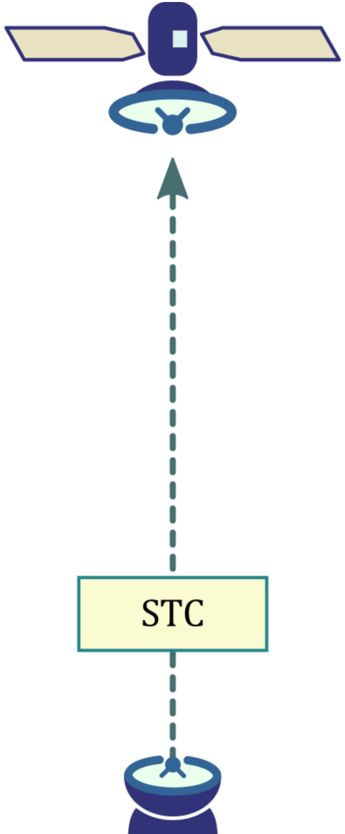
- Up-to-date data
  - ○ System resources (energy, memory, …)
  - ○ File sizes after compression
  - → Improve system efficiency

- Increased reactivity
  - ○ Recent weather forecasts (TC) or onboard detection
  - → Decrease rejection rate
  - → Reduce response delays



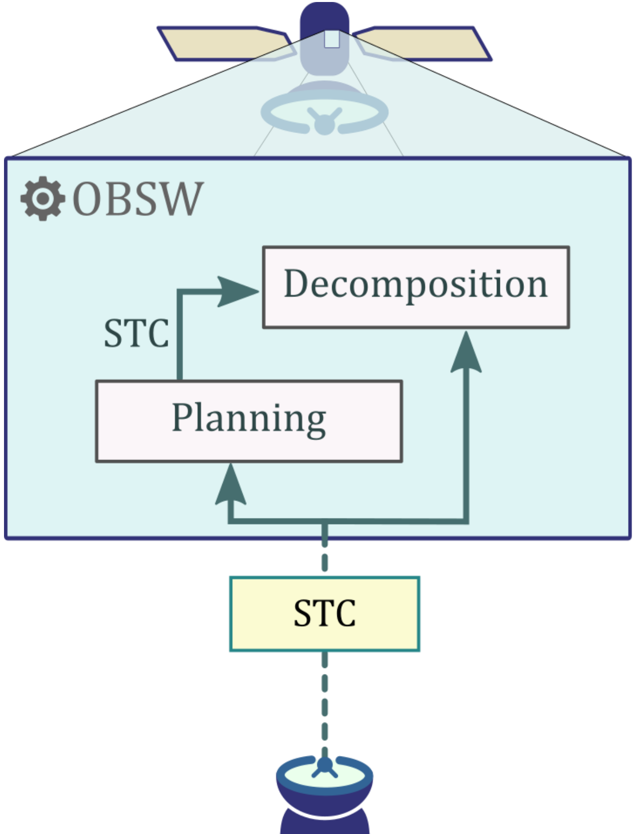
Synthetic Telecommand (STC)

- Macro command



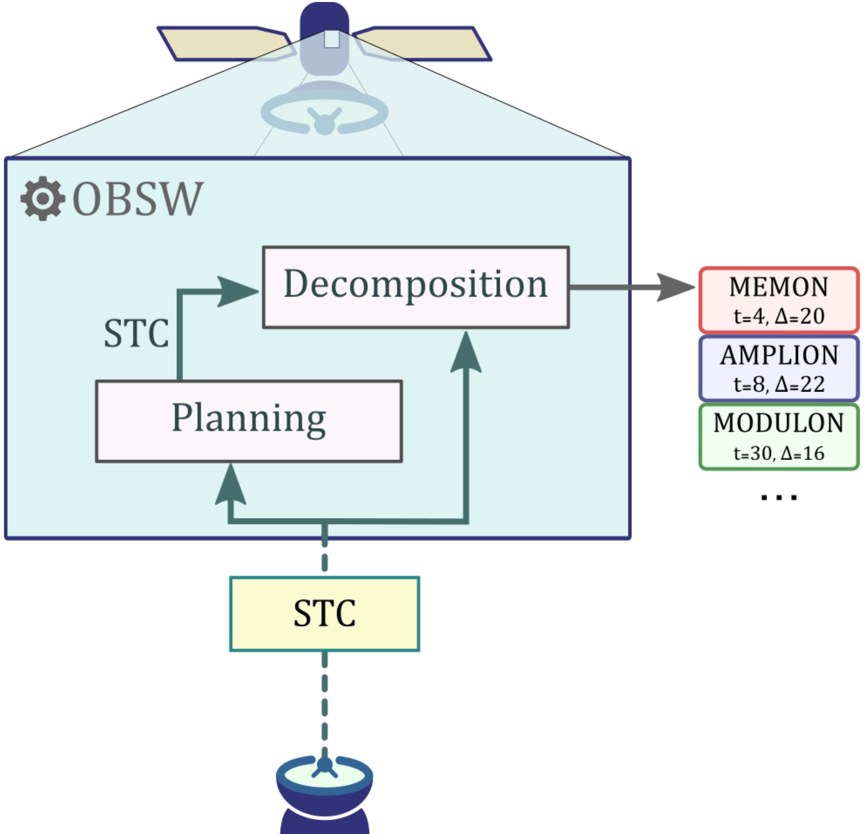
Synthetic Telecommand (STC)

- Macro command
- Decomposed on board



Synthetic Telecommand (STC)

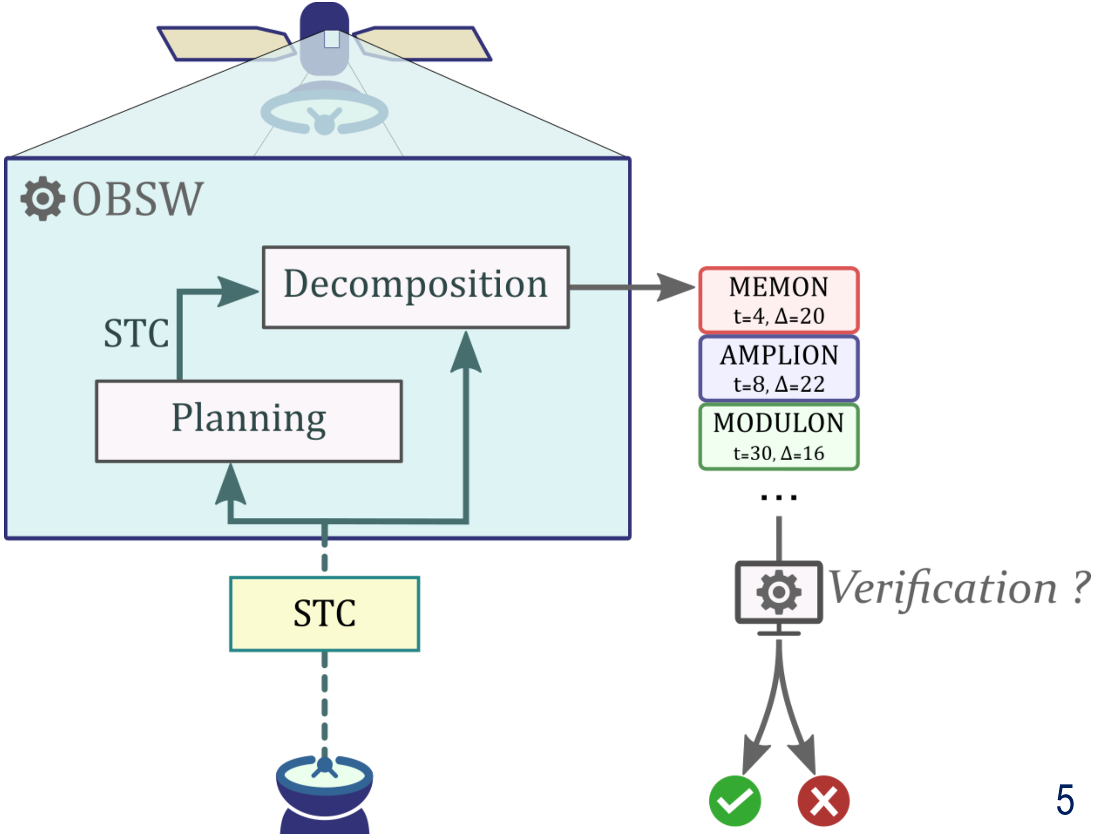
- Macro command
- Decomposed on board



Autonomy

Synthetic Telecommand (STC)

- Macro command
- Decomposed on board
- Need to be validated

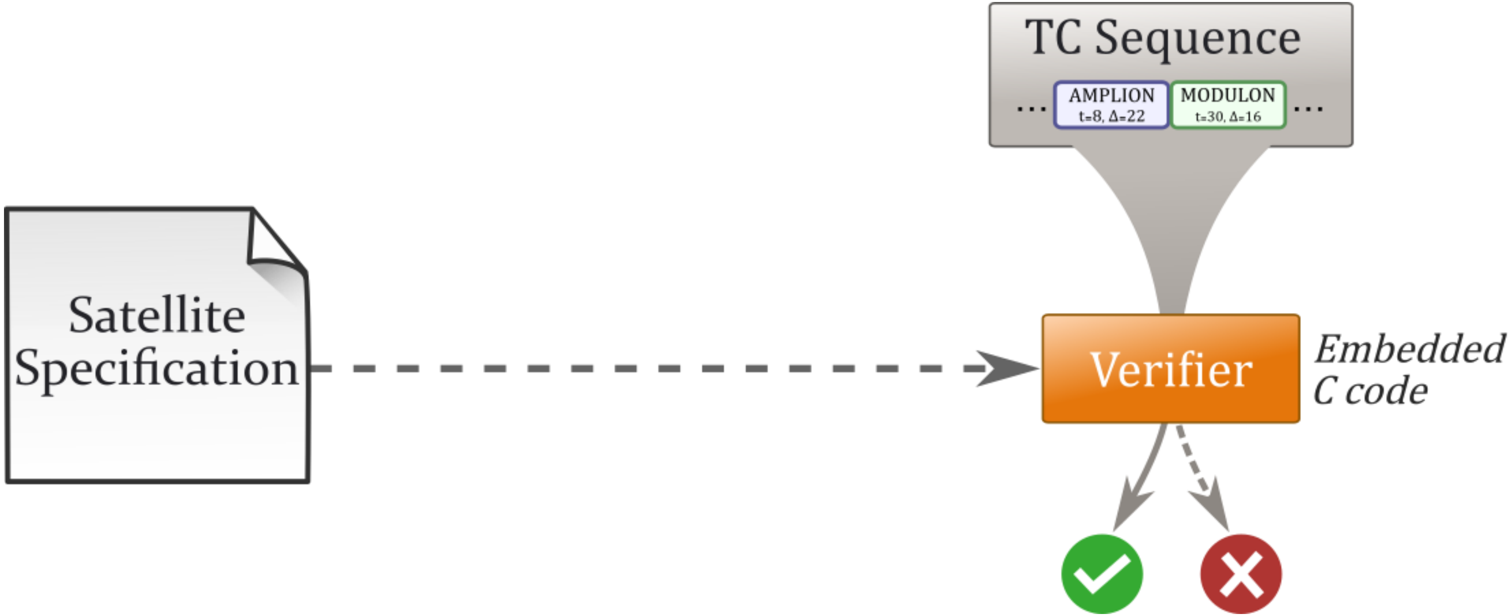


Summary

- Introduction
  - Context
  - Autonomy
- Formal Approach
  - Specification
  - Formal Model
- Experimentation
  - Framework
  - Prototype
- Conclusion

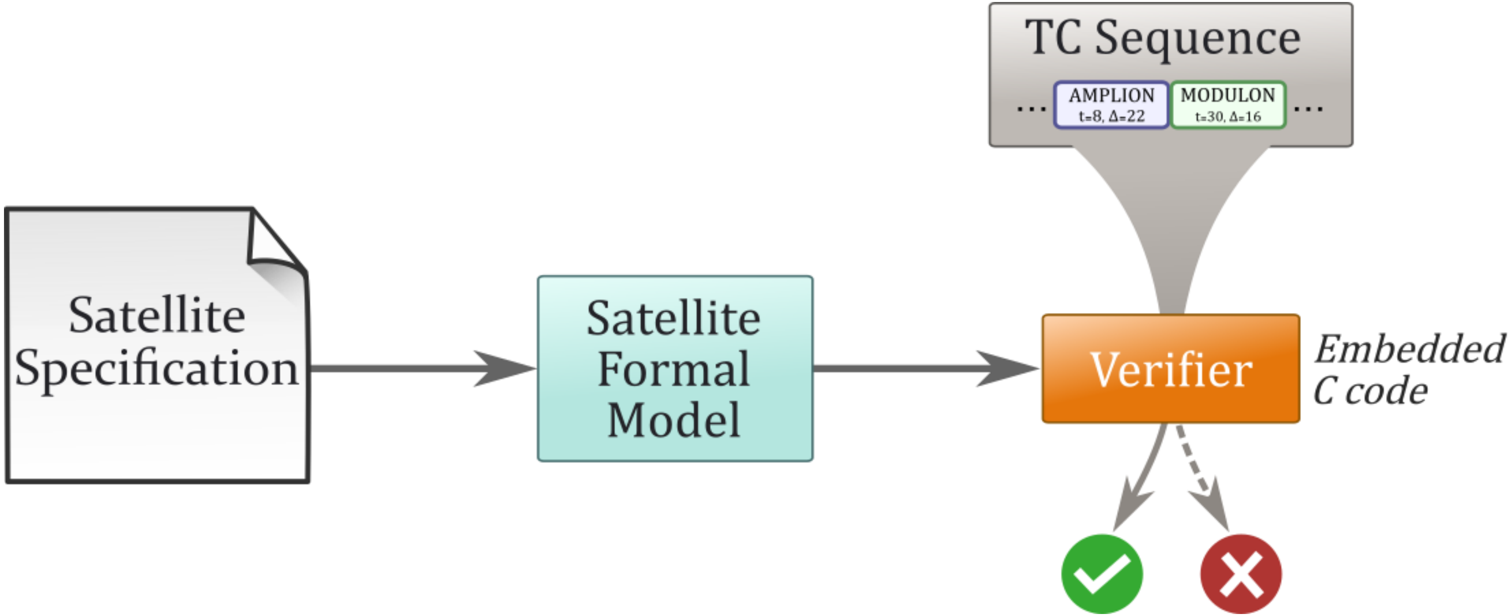
Formal approach

Decomposition verification = TC sequence verification



Formal approach

Decomposition verification = TC sequence verification



Specification

- Telecommand

  Effect & equipement / function targeted

- Durations

  Relative date

- Constraints

  Target & expected state

Satellite specification extract (SPOT-like)

REQ_DOWN_02 - TC **MODULON:**
Switch modulator to **ON.**
Modulator is **ON** after <DURATION_MODULON>
Initial condition: **Amplifier** is **ON**

Specification

- ➢ Telecommand

  Effect & equipement / function targeted

- ➢ Durations

  Relative date, variable durations

- ➢ Constraints

  Target & expected state

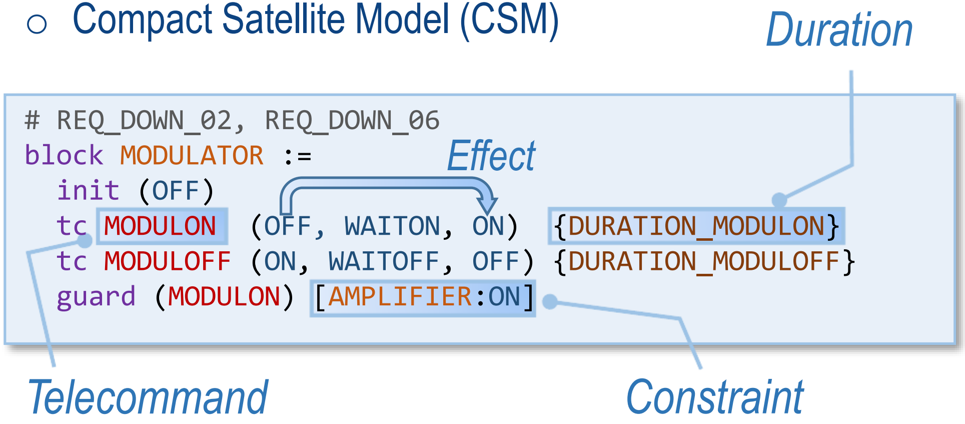
Satellite specification extract (SPOT-like)

REQ_DOWN_02 - TC **MODULON:**
Switch modulator to **ON.**
Modulator is **ON** after <DURATION_MODULON>
Initial condition: **Amplifier** is **ON**

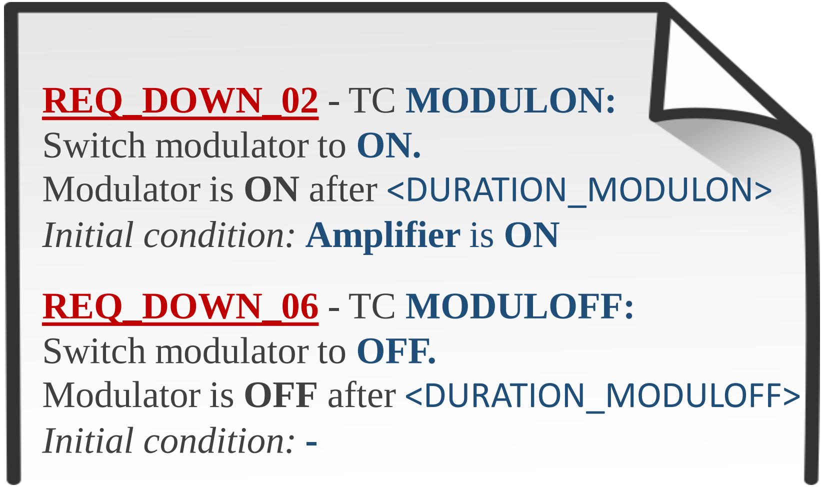
REQ_IMG_05 - TC **IMAGING:**
...
Imaging **STARTS** after <DURATION_IMAGING>
Imaging **STOPS** after <DELTA_MODULON>
...

Formal model

- Compact Satellite Model (CSM)



```
# REQ_DOWN_02, REQ_DOWN_06
block MODULATOR :=
  init (OFF)
  tc MODULON  (OFF, WAITON, ON)  {DURATION_MODULON}
  tc MODULOFF (ON, WAITOFF, OFF) {DURATION_MODULOFF}
  guard (MODULON) [AMPLIFIER:ON]
```

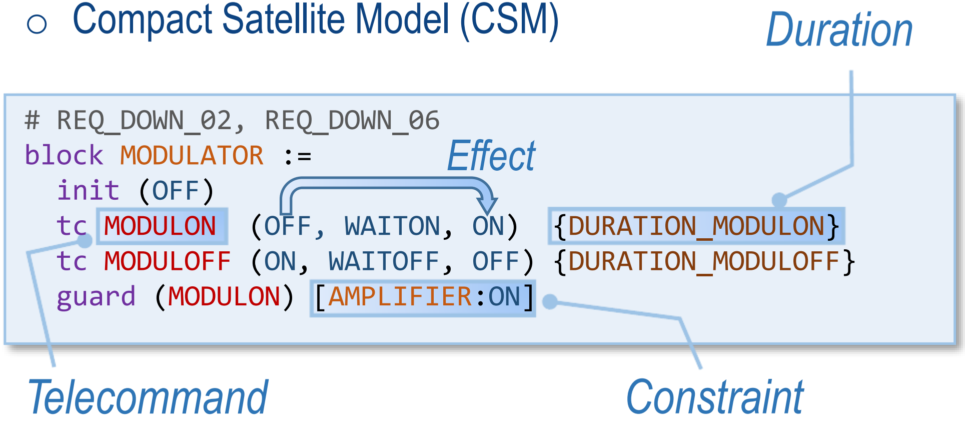


REQ_DOWN_02 - TC **MODULON:**
Switch modulator to **ON.**
Modulator is **ON** after <DURATION_MODULON>
Initial condition: **Amplifier** is **ON**

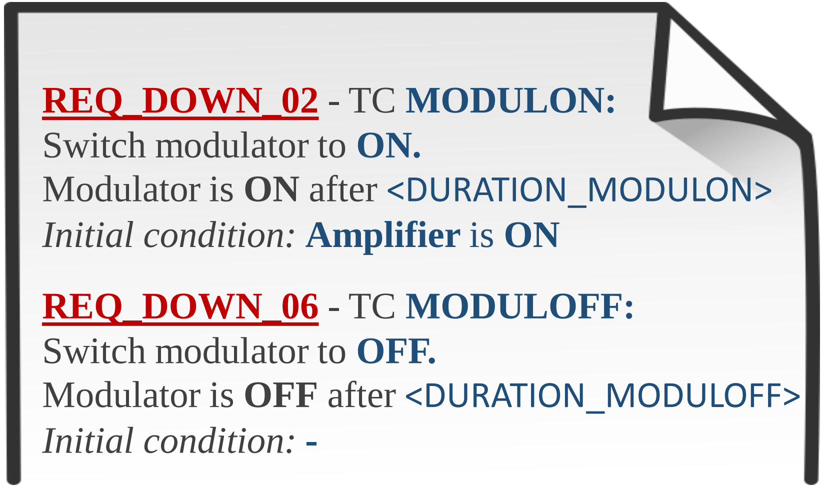
REQ_DOWN_06 - TC **MODULOFF:**
Switch modulator to **OFF.**
Modulator is **OFF** after <DURATION_MODULOFF>
Initial condition: **-**

Formal model

○ Compact Satellite Model (CSM)



```
# REQ_DOWN_02, REQ_DOWN_06
block MODULATOR :=
  init (OFF)
  tc MODULON  (OFF, WAITON, ON)  {DURATION_MODULON}
  tc MODULOFF (ON, WAITOFF, OFF) {DURATION_MODULOFF}
  guard (MODULON) [AMPLIFIER:ON]
```



REQ_DOWN_02 - TC **MODULON:**
Switch modulator to **ON.**
Modulator is **ON** after <DURATION_MODULON>
Initial condition: **Amplifier** is **ON**

REQ_DOWN_06 - TC **MODULOFF:**
Switch modulator to **OFF.**
Modulator is **OFF** after <DURATION_MODULOFF>
Initial condition: -

- ➢ *Domain Specific Language (DSL)*
- ✓ *Close to specification (with traceability)*
- ✓ *Compactness and Modularity*
- ✓ *Code and graph generation*

Formal model

- Compact Satellite Model (CSM)

```
# REQ_DOWN_02, REQ_DOWN_06
block MODULATOR :=
  init (OFF)
  tc MODULON  (OFF, WAITON, ON)  {DURATION_MODULON}
  tc MODULOFF (ON, WAITOFF, OFF) {DURATION_MODULOFF}
  guard (MODULON) [AMPLIFIER:ON]
```

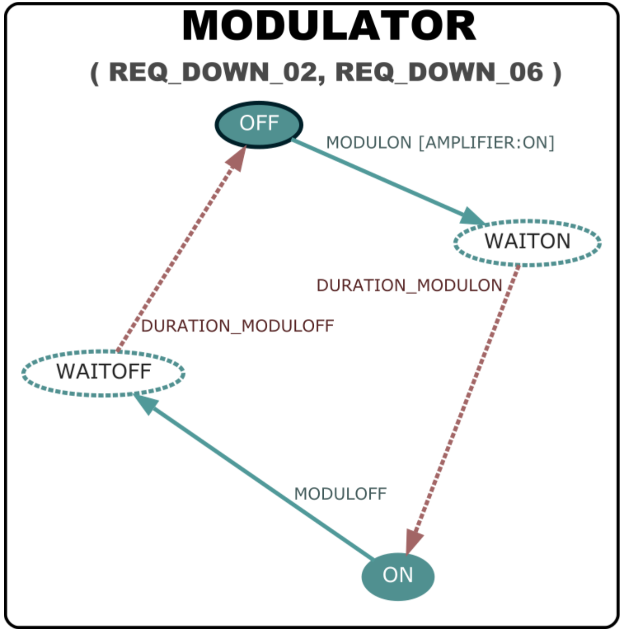
➢ *Domain Specific Language (DSL)*

✓ *Close to specification (with traceability)*

✓ *Compactness and Modularity*

✓ *Code and graph generation*

- Ad hoc automata formalism

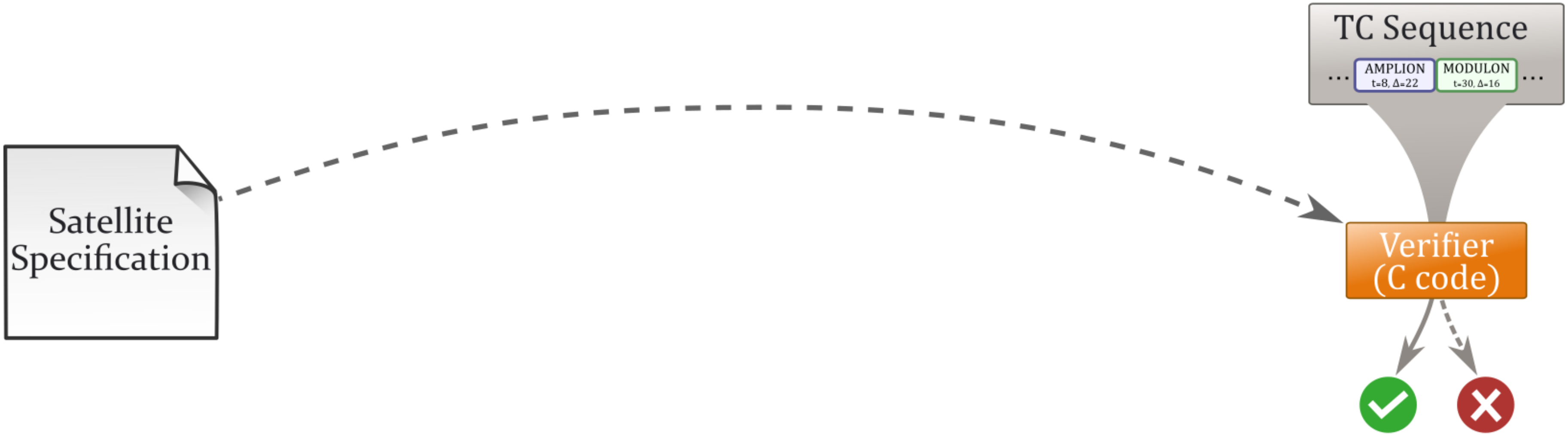


✓ *Modularity* ✓ *Traceability*

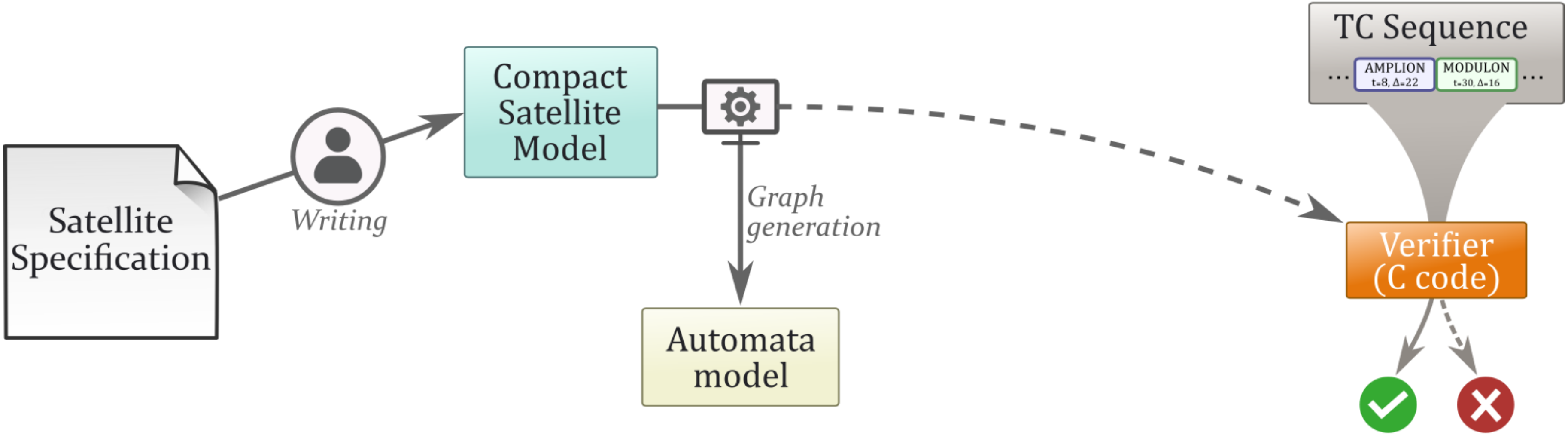
Summary

- Introduction
  - Context
  - Autonomy
- Formal Approach
  - Specification
  - Formal Model
- Experimentation
  - Framework
  - Prototype
- Conclusion

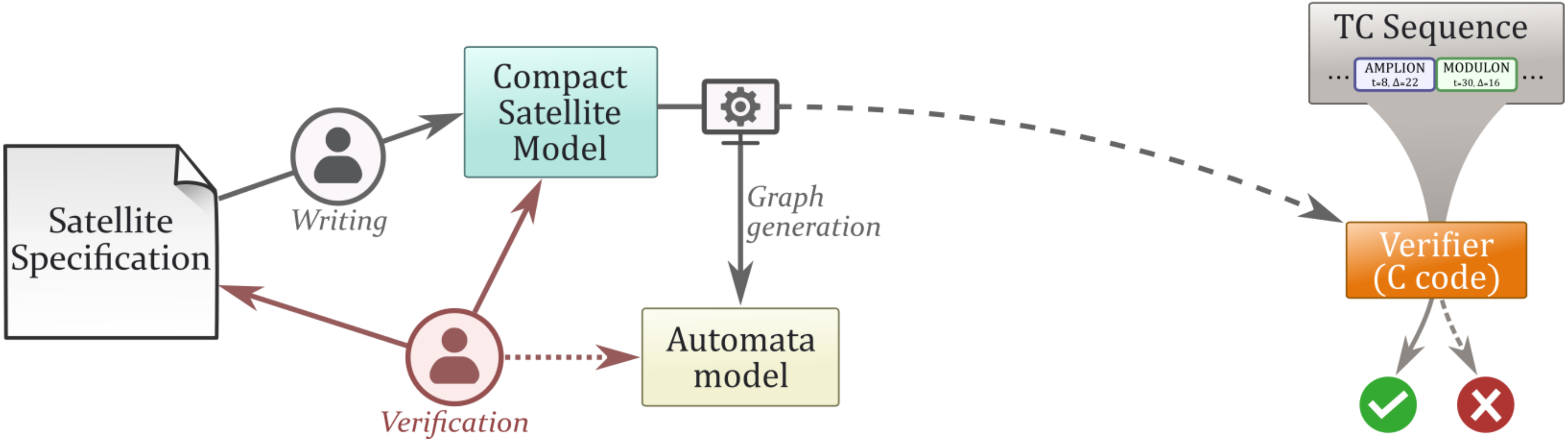
Framework


TC Sequence
…
AMPLION
t=8, Δ=22
MODULON
t=30, Δ=16
…
Satellite Specification
Verifier (C code)

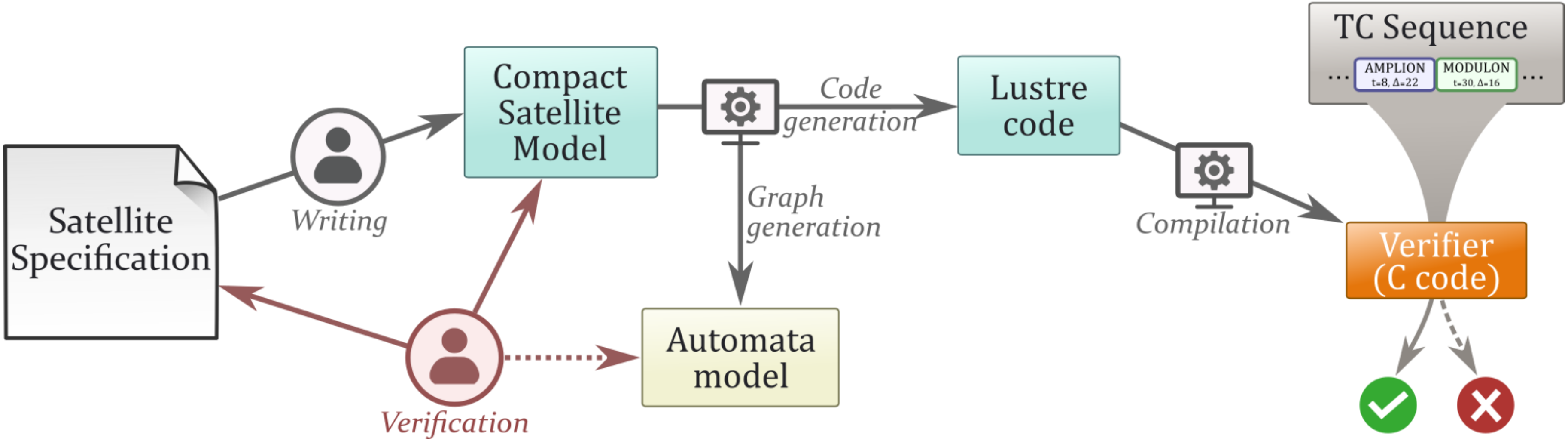
Framework


TC Sequence
…
AMPLION
t=8, Δ=22
MODULON
t=30, Δ=16
…
Satellite Specification
Writing
Compact Satellite Model
Graph generation
Automata model
Verifier (C code)

Framework

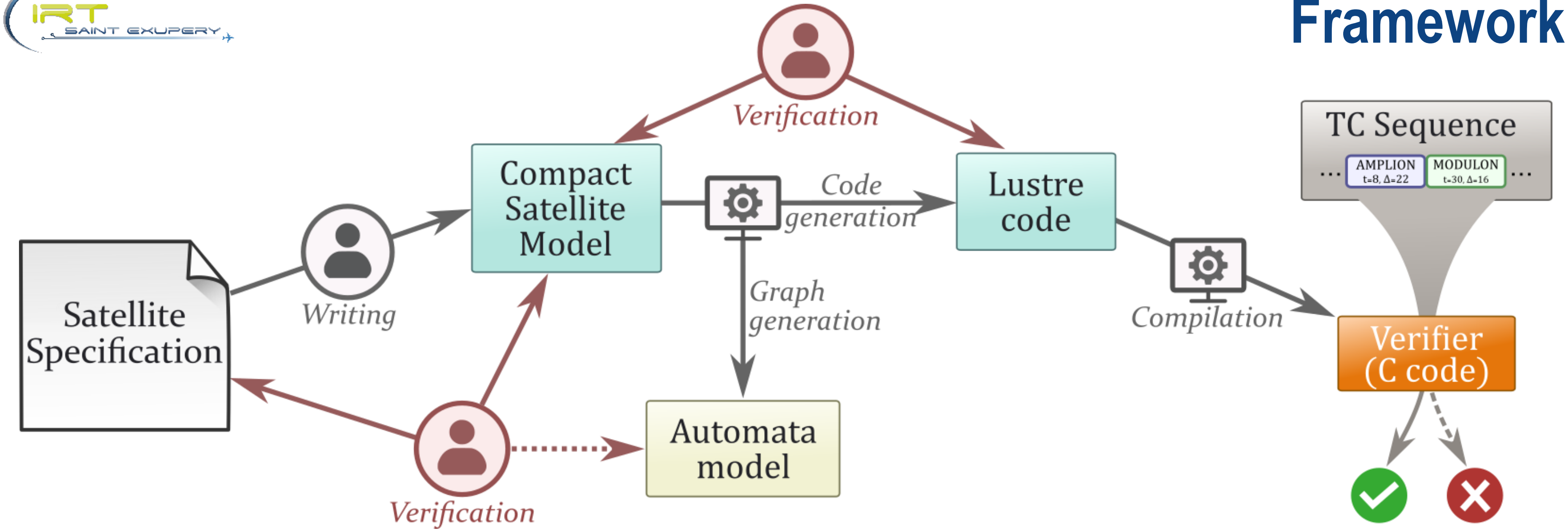

Satellite Specification
Writing
Compact Satellite Model
Graph generation
Automata model
Verification
TC Sequence
… AMPLION t=8, Δ=22
MODULON t=30, Δ=16 …
Verifier (C code)

Framework



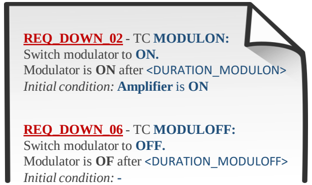
L U S T R E

Framework



L
U
S
T
R
E

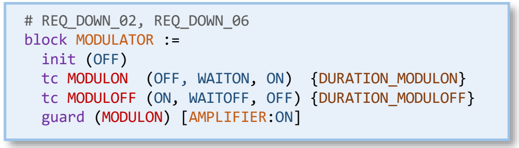
✓Same abstraction level



REQ_DOWN_02 - TC **MODULON:**
Switch modulator to **ON.**
Modulator is **ON** after <DURATION_MODULON>
Initial condition: **Amplifier** is **ON**

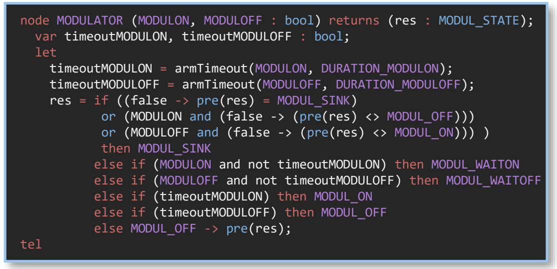
REQ_DOWN_06 - TC **MODULOFF:**
Switch modulator to **OFF.**
Modulator is **OF** after <DURATION_MODULOFF>
Initial condition: -

Specification



```
# REQ_DOWN_02, REQ_DOWN_06
block MODULATOR :=
  init (OFF)
  tc MODULON  (OFF, WAITON, ON)  {DURATION_MODULON}
  tc MODULOFF (ON, WAITOFF, OFF) {DURATION_MODULOFF}
  guard (MODULON) [AMPLIFIER:ON]
```

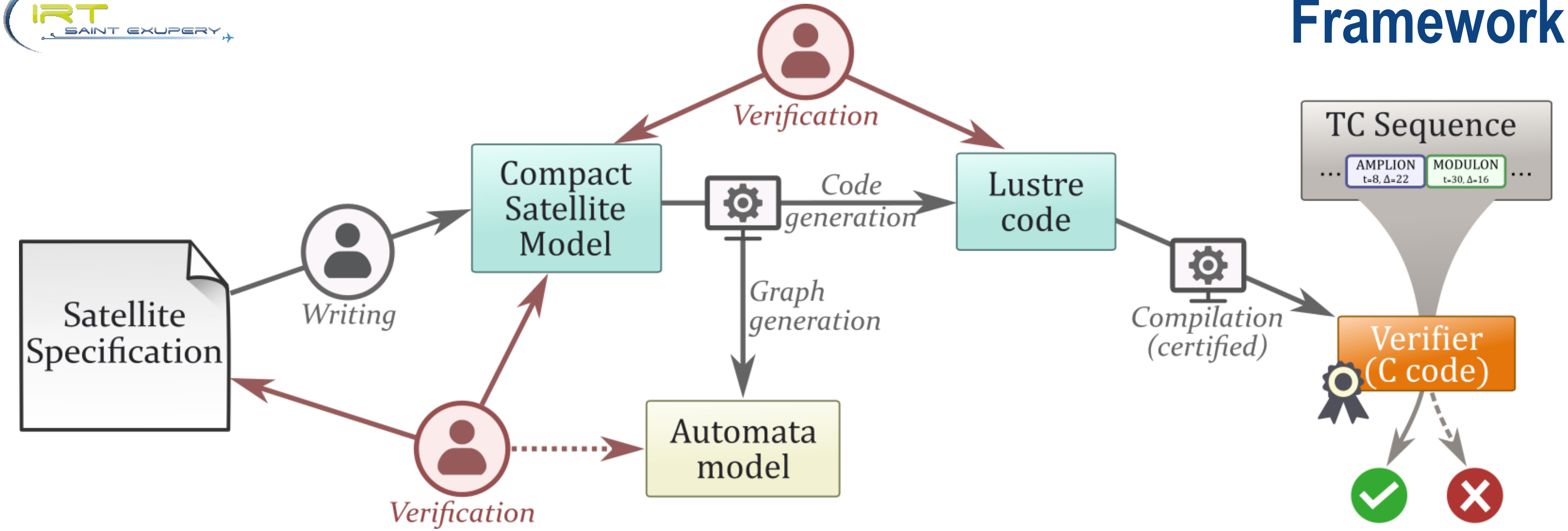
CSM



```
node MODULATOR (MODULON, MODULOFF : bool) returns (res : MODUL_STATE);
  var timeoutMODULON, timeoutMODULOFF : bool;
  let
    timeoutMODULON = armTimeout(MODULON, DURATION_MODULON);
    timeoutMODULOFF = armTimeout(MODULOFF, DURATION_MODULOFF);
    res = if ((false -> pre(res) = MODUL_SINK)
           or (MODULON and (false -> (pre(res) <> MODUL_OFF)))
           or (MODULOFF and (false -> (pre(res) <> MODUL_ON))) )
           then MODUL_SINK
          else if (MODULON and not timeoutMODULON) then MODUL_WAITON
          else if (MODULOFF and not timeoutMODULOFF) then MODUL_WAITOFF
          else if (timeoutMODULON) then MODUL_ON
          else if (timeoutMODULOFF) then MODUL_OFF
          else MODUL_OFF -> pre(res);
tel
```

Lustre

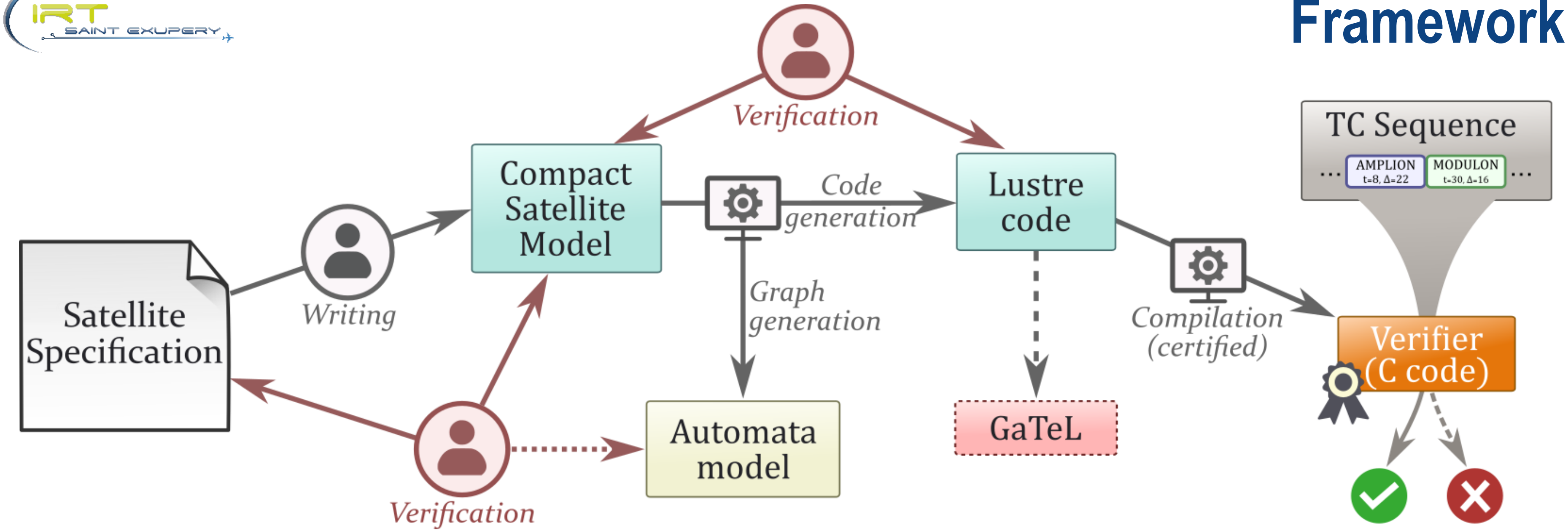
Framework



LUSTRE

- ✓Same abstraction level
- ✓Sound (*Used in avionics, certifiable DO-178A*)

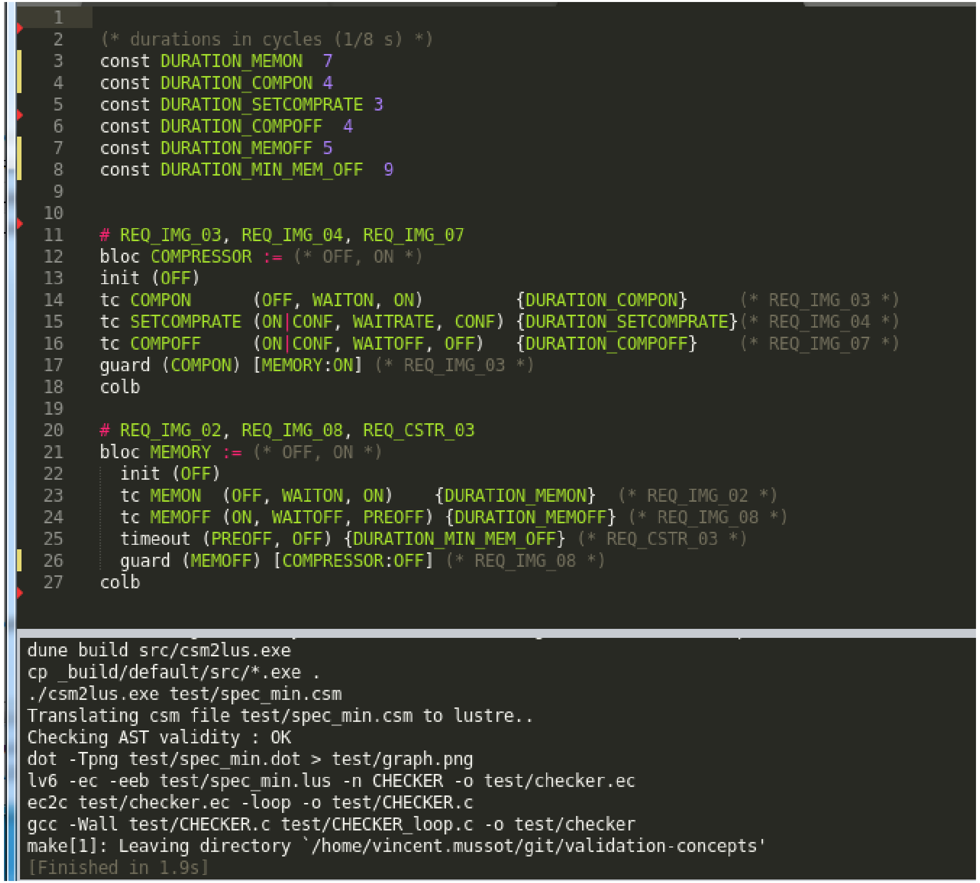
Framework



L U S T R E

- ✓Same abstraction level
- ✓Sound (*Used in avionics, certifiable DO-178A*)
- ✓Available tools (Scade, Prover, Kind2, GaTeL...)

Prototype

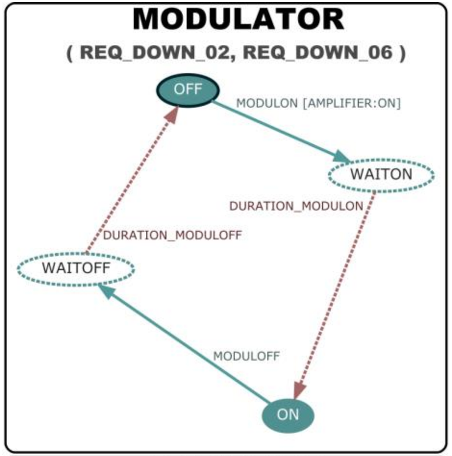


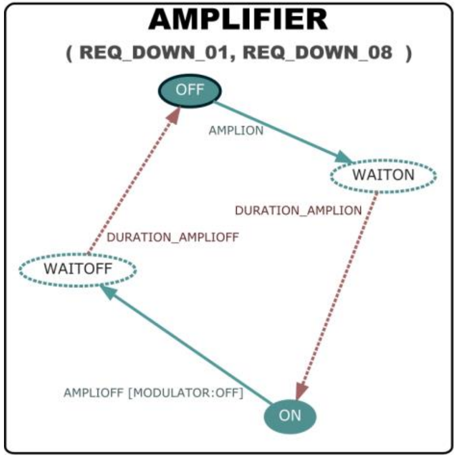
```
(* durations in cycles (1/8 s) *)
const DURATION_MEMON  7
const DURATION_COMPON 4
const DURATION_SETCOMPRATE 3
const DURATION_COMPOFF  4
const DURATION_MEMOFF 5
const DURATION_MIN_MEM_OFF  9


# REQ_IMG_03, REQ_IMG_04, REQ_IMG_07
bloc COMPRESSOR := (* OFF, ON *)
init (OFF)
tc COMPON      (OFF, WAITON, ON)         {DURATION_COMPON}     (* REQ_IMG_03 *)
tc SETCOMPRATE (ON|CONF, WAITRATE, CONF) {DURATION_SETCOMPRATE}(* REQ_IMG_04 *)
tc COMPOFF     (ON|CONF, WAITOFF, OFF)   {DURATION_COMPOFF}    (* REQ_IMG_07 *)
guard (COMPON) [MEMORY:ON] (* REQ_IMG_03 *)
colb

# REQ_IMG_02, REQ_IMG_08, REQ_CSTR_03
bloc MEMORY := (* OFF, ON *)
  init (OFF)
  tc MEMON  (OFF, WAITON, ON)    {DURATION_MEMON}  (* REQ_IMG_02 *)
  tc MEMOFF (ON, WAITOFF, PREOFF) {DURATION_MEMOFF} (* REQ_IMG_08 *)
  timeout (PREOFF, OFF) {DURATION_MIN_MEM_OFF} (* REQ_CSTR_03 *)
  guard (MEMOFF) [COMPRESSOR:OFF] (* REQ_IMG_08 *)
colb
```

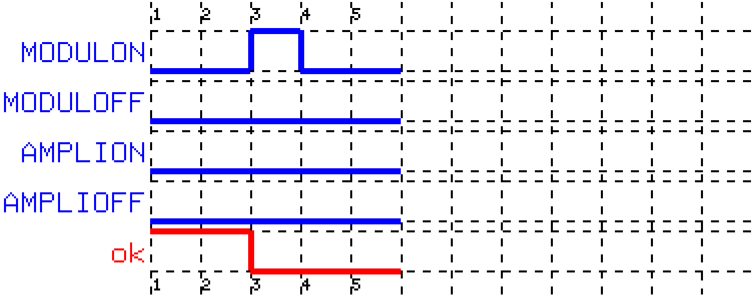
```
dune build src/csm2lus.exe
cp _build/default/src/*.exe .
./csm2lus.exe test/spec_min.csm
Translating csm file test/spec_min.csm to lustre..
Checking AST validity : OK
dot -Tpng test/spec_min.dot > test/graph.png
lv6 -ec -eeb test/spec_min.lus -n CHECKER -o test/checker.ec
ec2c test/checker.ec -loop -o test/CHECKER.c
gcc -Wall test/CHECKER.c test/CHECKER_loop.c -o test/checker
make[1]: Leaving directory `/home/vincent.mussot/git/validation-concepts'
[Finished in 1.9s]
```



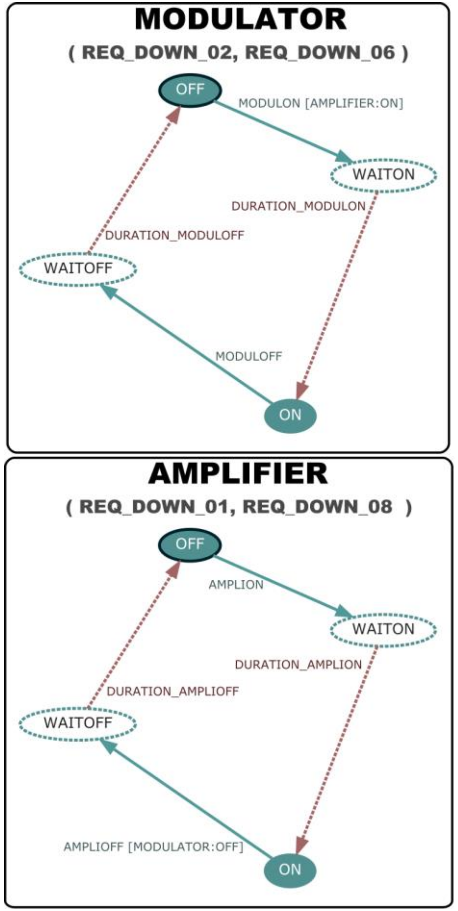


Prototype

- *Lustre Simulation*

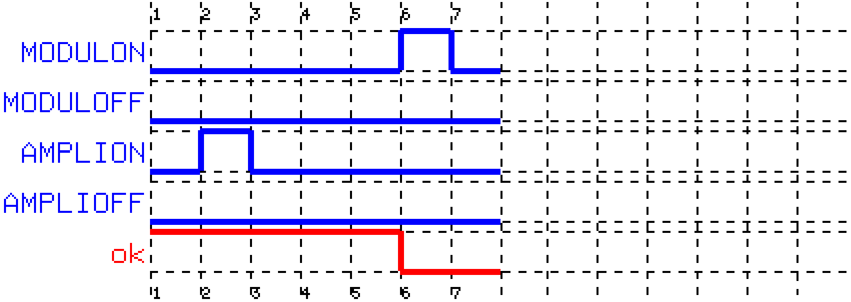


➔MODULON when Amplifier is Not ON

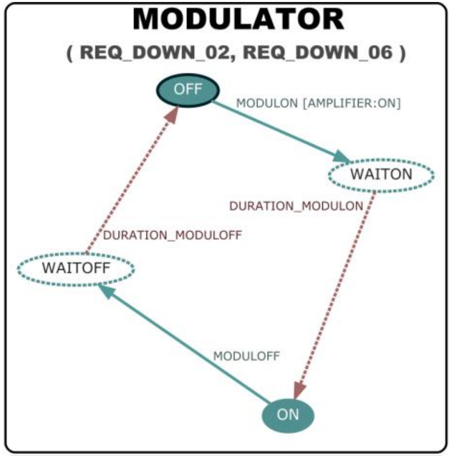


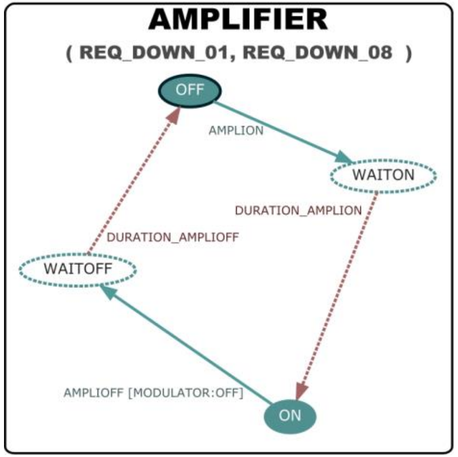
Prototype

- ○ *Lustre Simulation*



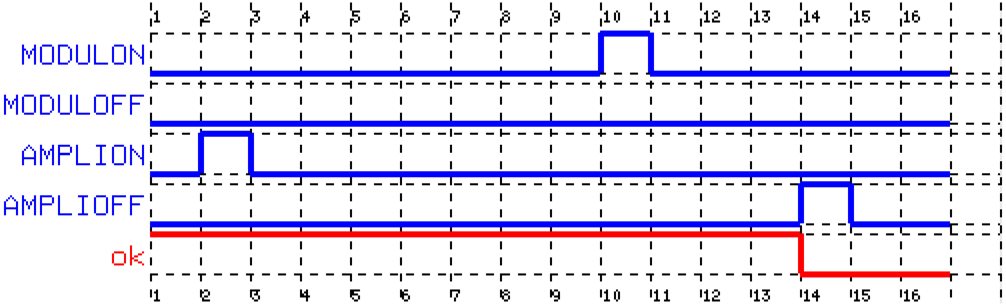
➔MODULON before AMPLION timeout



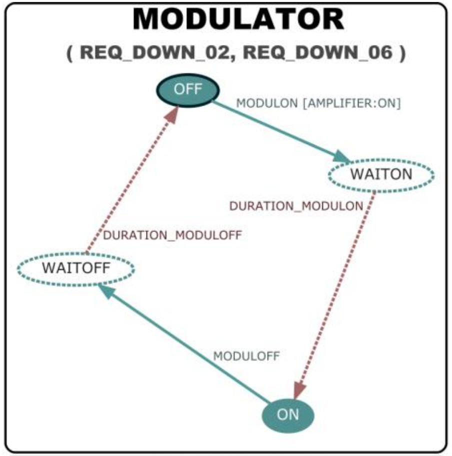


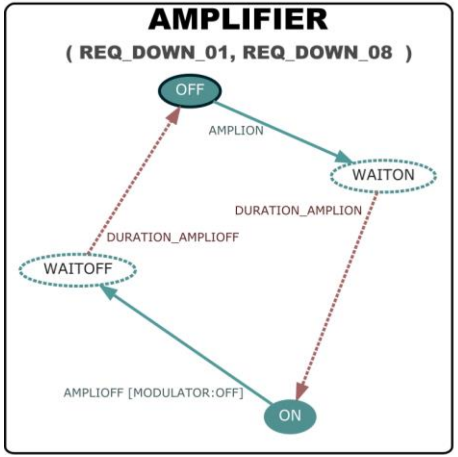
Prototype

- ○ *Lustre Simulation*

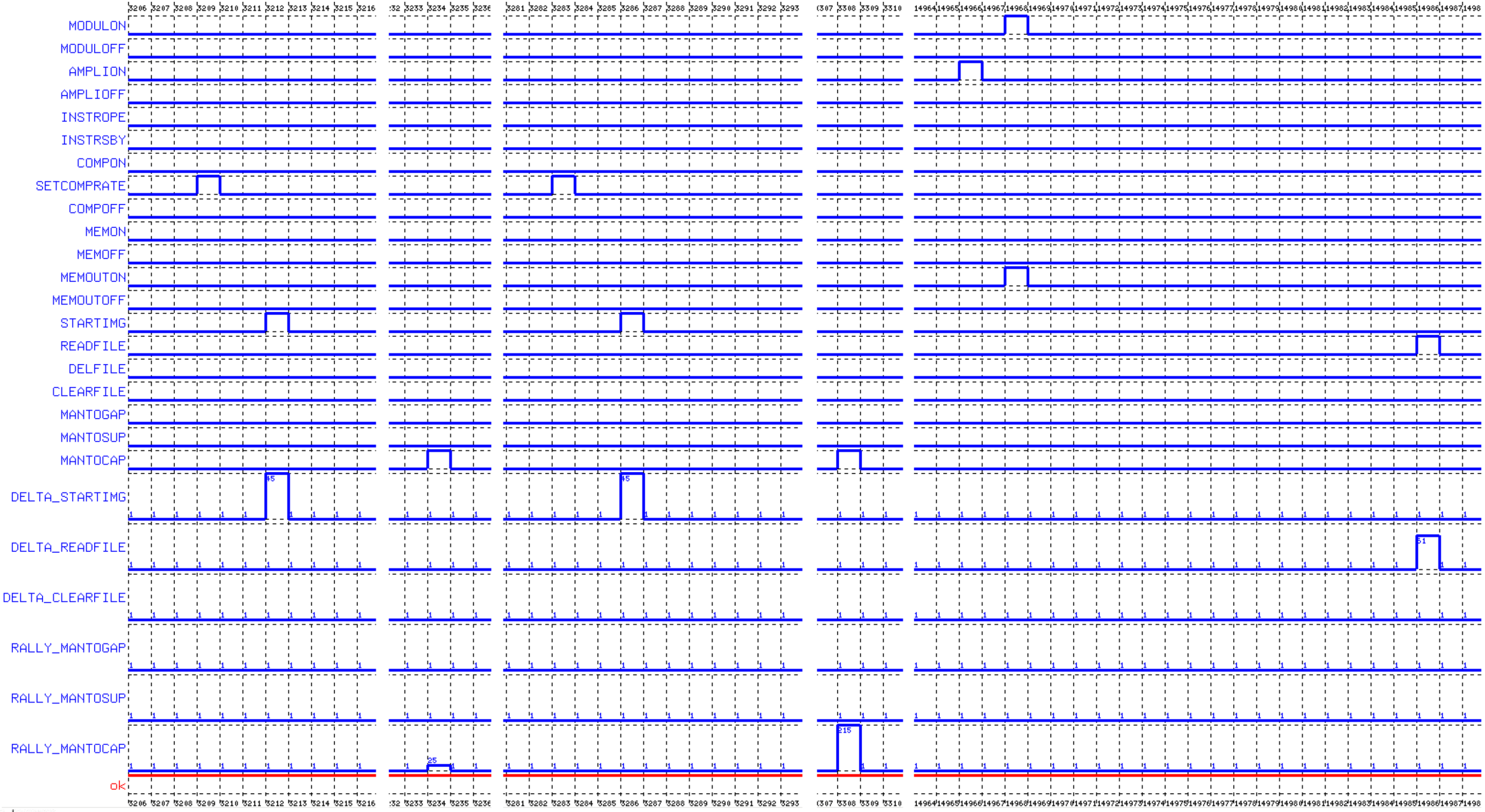


➔AMPLIOFF when Modulator is Not OFF





Prototype



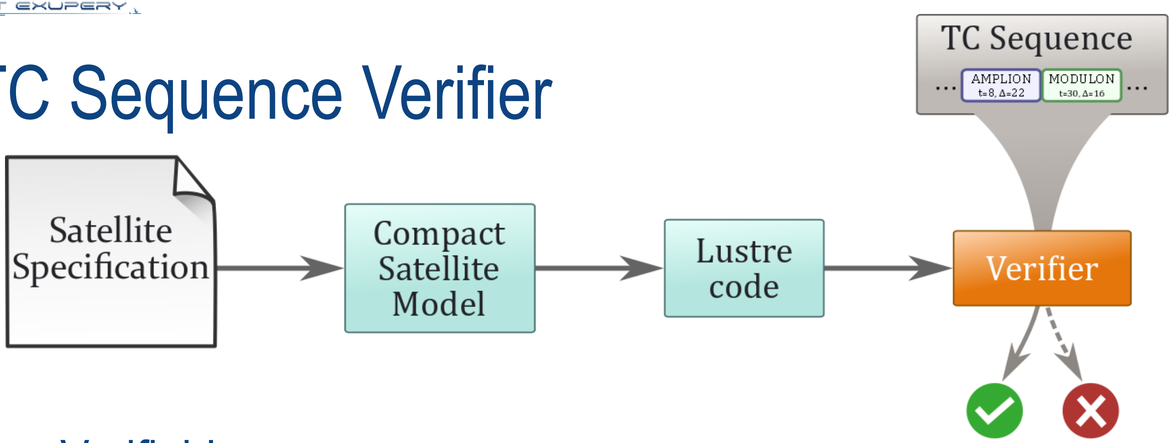
Extract from Agata TM report

Summary

- Introduction
  - Context
  - Autonomy
- Formal Approach
  - Specification
  - Formal Model
- Experimentation
  - Framework
  - Prototype
- Conclusion

Conclusion

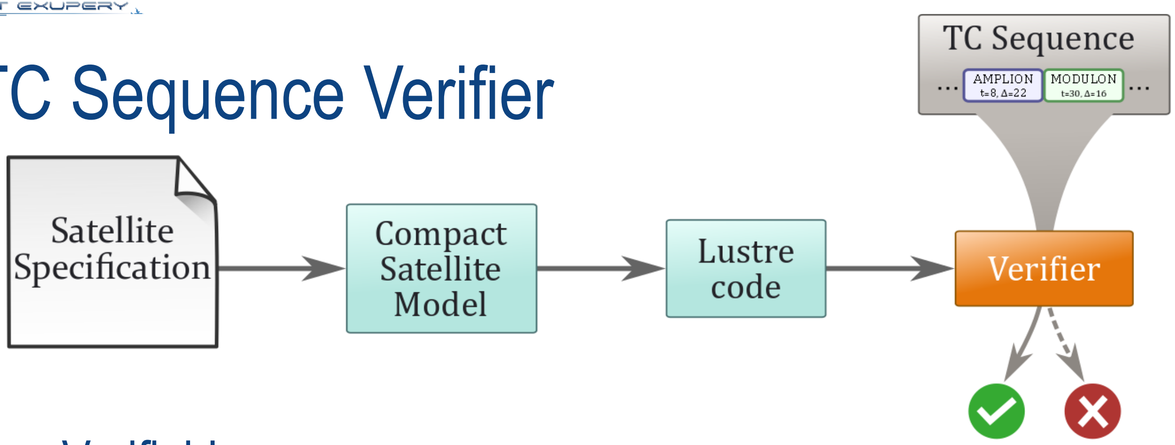
A TC Sequence Verifier



- Verifiable
- Based on formal design
- Minimal human input (code generation)
- Embeddable (small)

Conclusion

A TC Sequence Verifier



- Verifiable
- Based on formal design
- Minimal human input (code generation)
- Embeddable (small)

Future work

- Test generation with GaTeL
- Event-based Lustre simulation