Code generation

C. Tanci, G. Tosti
Università di Perugia

Code generation rationale

- Code generation ease the task of writing repetitive code: ACS characteristic components have usually many repetitions of similar snippets in the code and in the configuration files to implement BACI properties and their configuration
  - ASTRI Telescope Control Unit consist of 352 control points:
    ~9000 lines of implementation code as ACS properties + methods
    ~1100 lines of CDB schema
    ~4700 lines of configuration file **without business logic**
- It reduces the number of lines of code that a developer needs to craft
- It allows to concentrate resources on higher levels of abstraction
- It reduces time from interface specification to testing

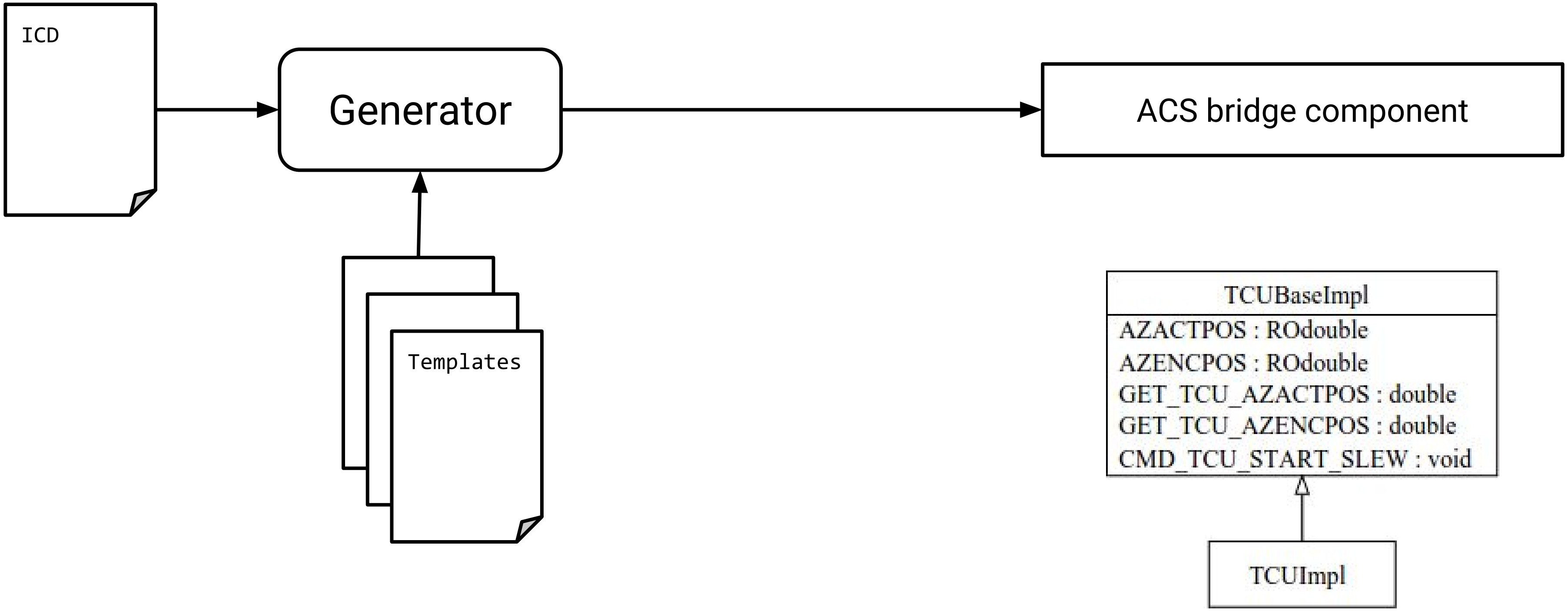
ALMA and people in CTA are also using code generation

- Troncoso, Nicolás, et al. "*A code generation framework for the ALMA common software.*" SPIE Astronomical Telescopes+ Instrumentation. International Society for Optics and Photonics, 2010.

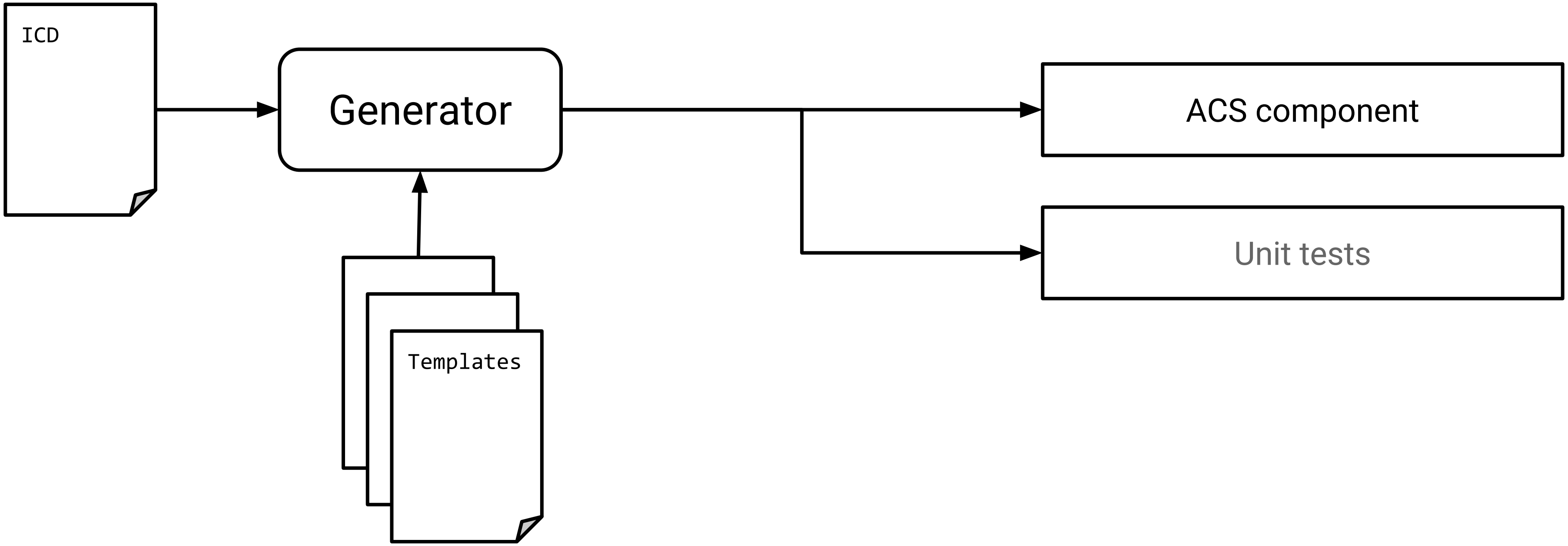
A few practical considerations

- Generated code must be kept isolated via extension or composition from manually developed logic code to limit the side effects of regeneration
- Code generation should be a step in a more general workflow of continuous integration process comprising code analysis, building, testing
- Code generated must be clean and humanly readable to allow easier debug activities. It should follow the same best practices and standards required for human written code
- We should try to not have an ICD for human consumption and one for generation unless one could be derived from the other
  - **Every piece of knowledge must have a single, unambiguous, authoritative representation within a system** DRY (Don't Repeat Yourself) Principle

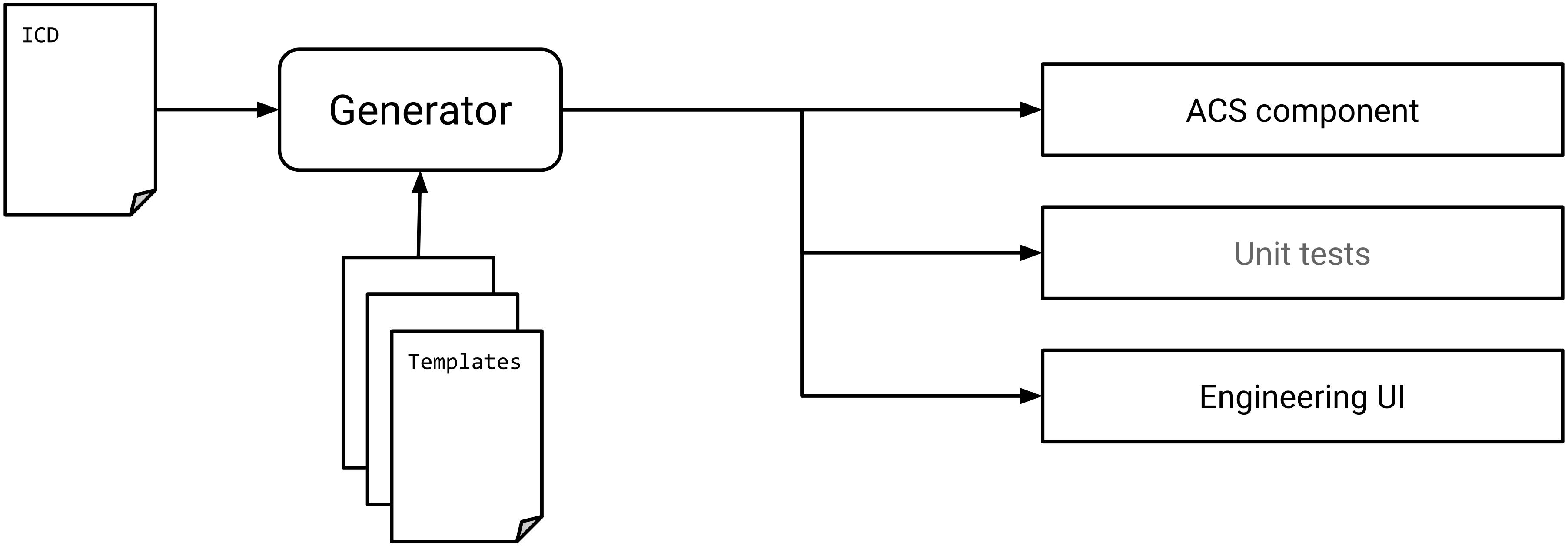
Code generation prototype output



Code generation prototype output



Code generation prototype output



ICD

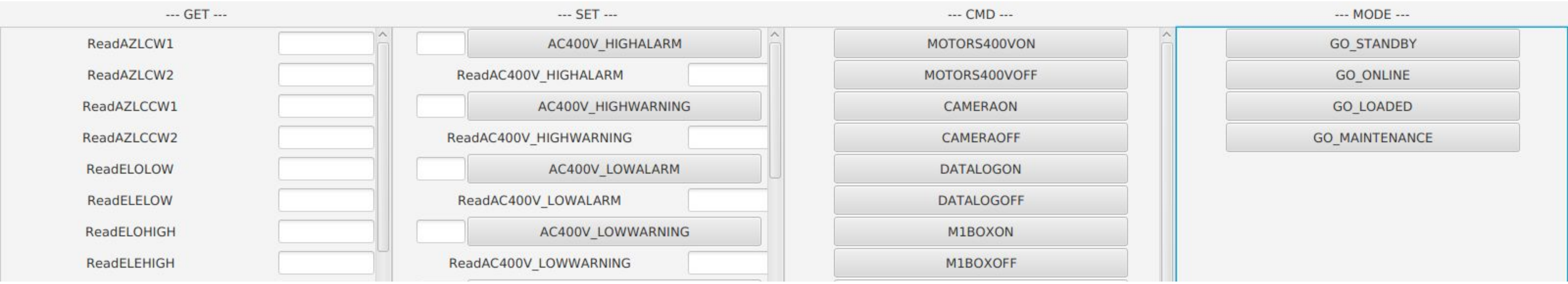
Generator

Templates

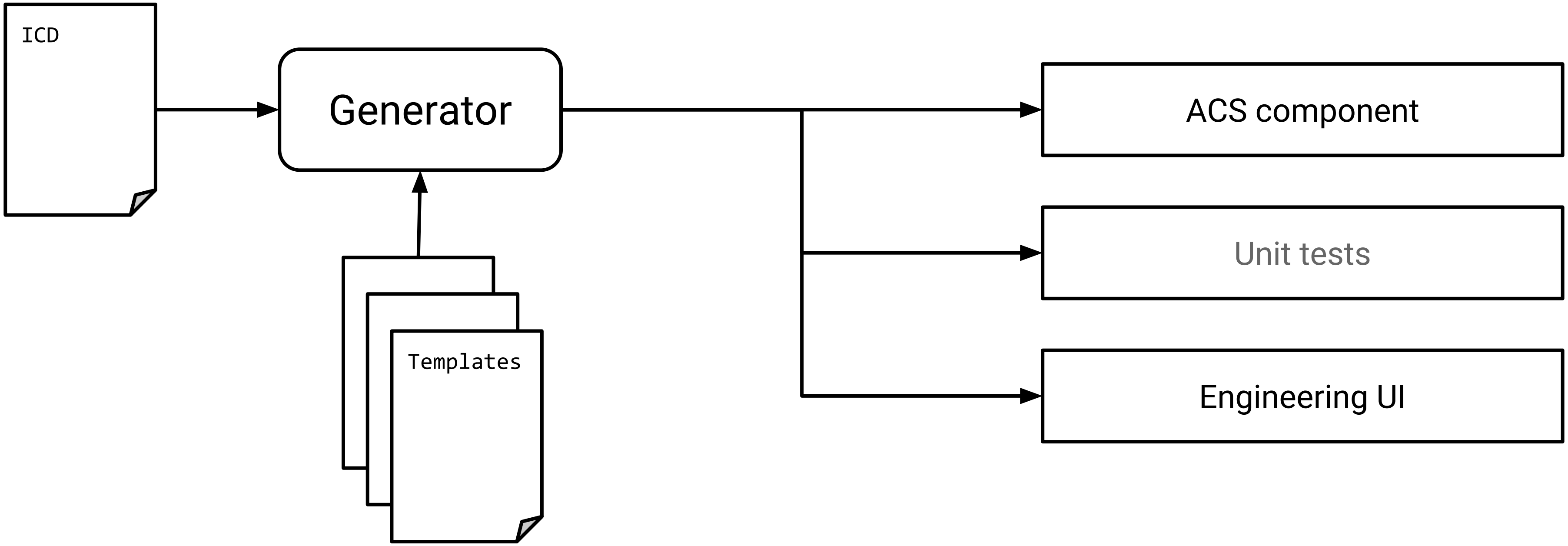
ACS component

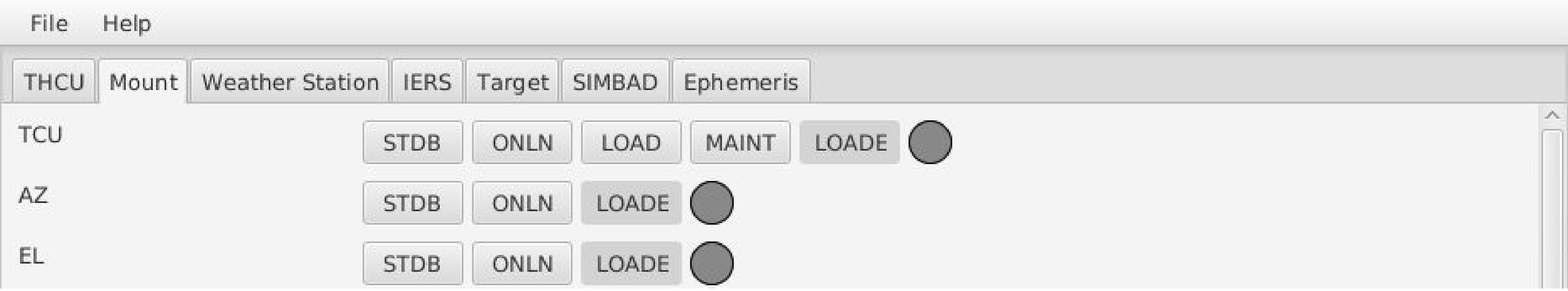
Unit tests

Engineering UI

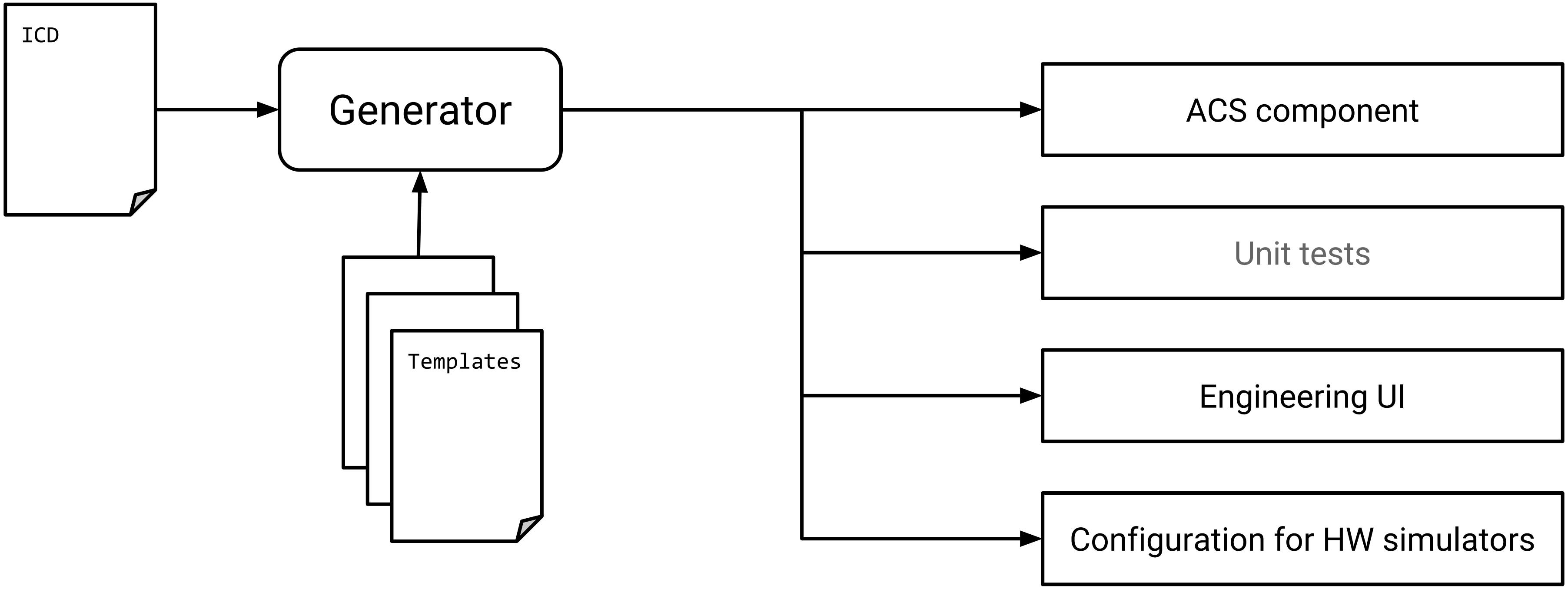


Code generation prototype output

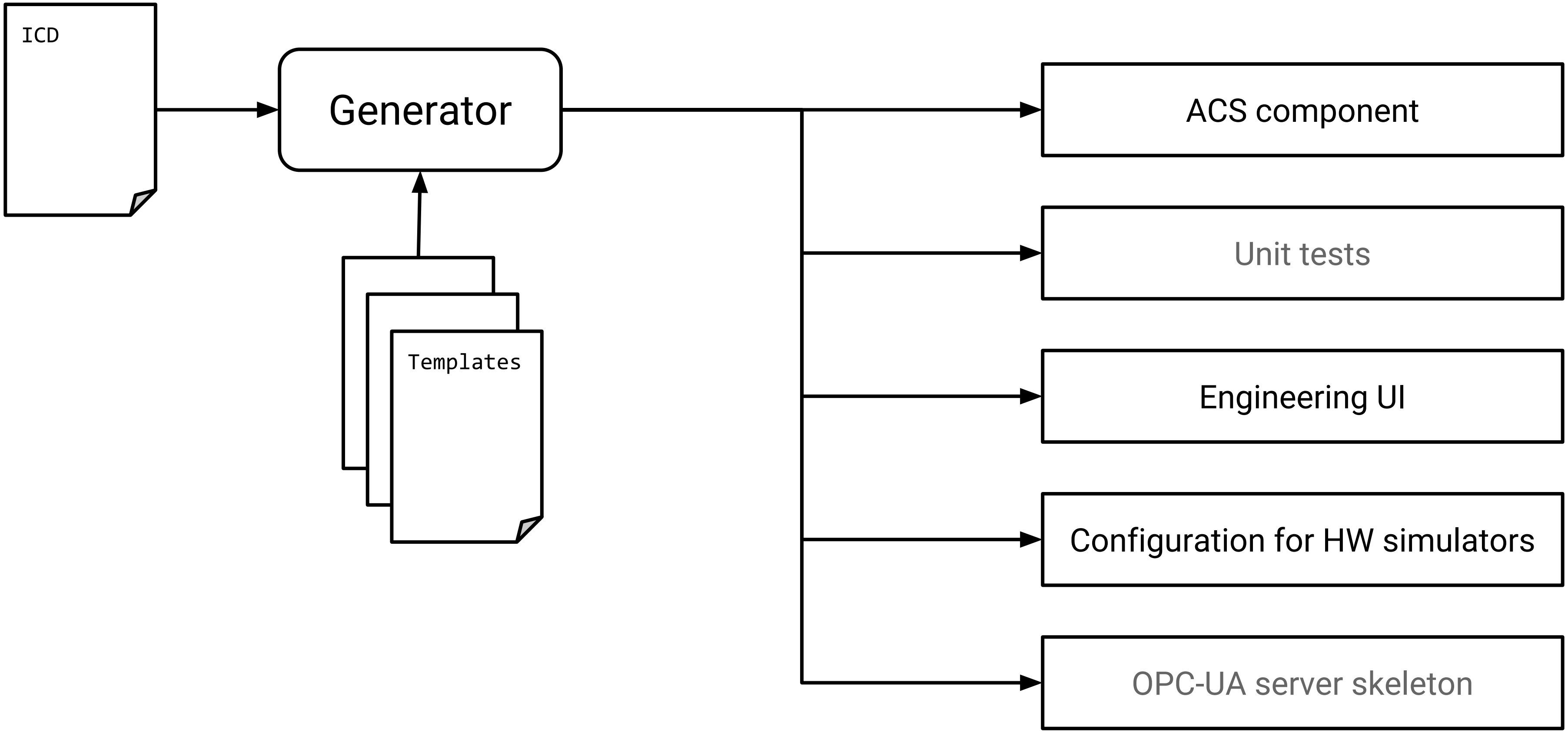




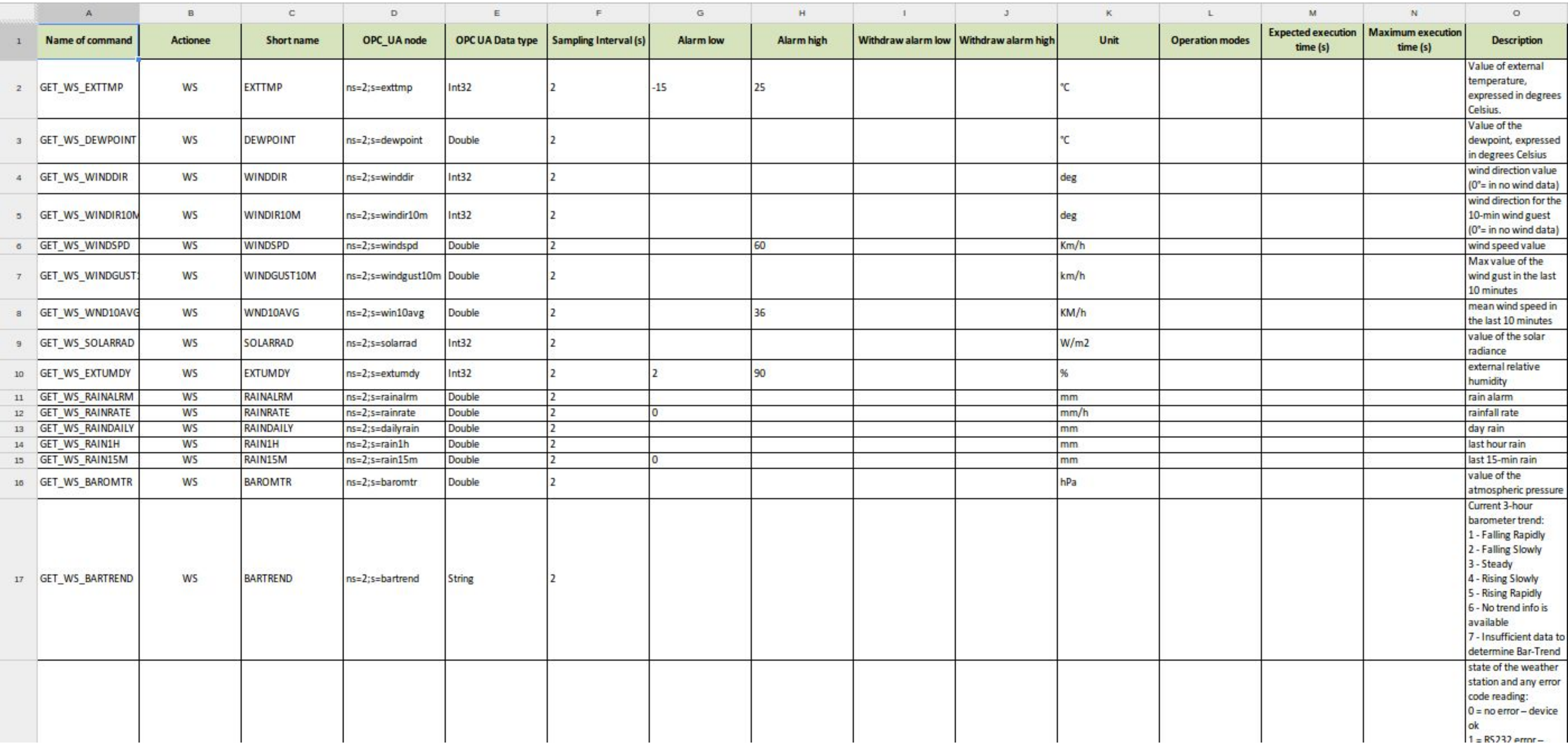
Code generation prototype output



Code generation prototype output



Code generation input



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**Name of command**	**Actionee**	**Short name**	**OPC_UA node**	**OPC UA Data type**	**Sampling Interval (s)**	**Alarm low**	**Alarm high**	**Withdraw alarm low**	**Withdraw alarm high**	**Unit**	**Operation modes**	**Expected execution time (s)**	**Maximum execution time (s)**	**Description**
2	GET_WS_EXTTMP	WS	EXTTMP	ns=2;s=exttmp	Int32	2	-15	25			°C				Value of external temperature, expressed in degrees Celsius.
3	GET_WS_DEWPOINT	WS	DEWPOINT	ns=2;s=dewpoint	Double	2					°C				Value of the dewpoint, expressed in degrees Celsius
4	GET_WS_WINDDIR	WS	WINDDIR	ns=2;s=winddir	Int32	2					deg				wind direction value (0°= in no wind data)
5	GET_WS_WINDIR10M	WS	WINDIR10M	ns=2;s=windir10m	Int32	2					deg				wind direction for the 10-min wind guest (0°= in no wind data)
6	GET_WS_WINDSPD	WS	WINDSPD	ns=2;s=windspd	Double	2		60			Km/h				wind speed value
7	GET_WS_WINDGUST1	WS	WINDGUST10M	ns=2;s=windgust10m	Double	2					km/h				Max value of the wind gust in the last 10 minutes
8	GET_WS_WND10AVG	WS	WND10AVG	ns=2;s=win10avg	Double	2		36			KM/h				mean wind speed in the last 10 minutes
9	GET_WS_SOLARRAD	WS	SOLARRAD	ns=2;s=solarrad	Int32	2					W/m2				value of the solar radiance
10	GET_WS_EXTUMDY	WS	EXTUMDY	ns=2;s=extumdy	Int32	2	2	90			%				external relative humidity
11	GET_WS_RAINALRM	WS	RAINALRM	ns=2;s=rainalrm	Double	2					mm				rain alarm
12	GET_WS_RAINRATE	WS	RAINRATE	ns=2;s=rainrate	Double	2	0				mm/h				rainfall rate
13	GET_WS_RAINDAILY	WS	RAINDAILY	ns=2;s=dailyrain	Double	2					mm				day rain
14	GET_WS_RAIN1H	WS	RAIN1H	ns=2;s=rain1h	Double	2					mm				last hour rain
15	GET_WS_RAIN15M	WS	RAIN15M	ns=2;s=rain15m	Double	2	0				mm				last 15-min rain
16	GET_WS_BAROMTR	WS	BAROMTR	ns=2;s=baromtr	Double	2					hPa				value of the atmospheric pressure
17	GET_WS_BARTREND	WS	BARTREND	ns=2;s=bartrend	String	2									Current 3-hour barometer trend: 1 - Falling Rapidly 2 - Falling Slowly 3 - Steady 4 - Rising Slowly 5 - Rising Rapidly 6 - No trend info is available 7 - Insufficient data to determine Bar-Trend
															state of the weather station and any error code reading: 0 = no error – device ok 1 = RS232 error –

An Excel file with all the monitor and control points

Prototype limitations

- It takes as input excel tables
- Unit tests and other potential useful outputs are not generated
- Input phase is not well decoupled from output generation
- It needs documentation, tests, examples

Developments

- Some work on extending the tool
  - Support for arrays in progress
  - Templates for generated code must be reviewed and validated
- Input and output
  - Input from official ICDs? A more strict metadata description format? Directly from an OPC-UA server?
  - Optimized output with support for ACS methods, alarms - is there demand?
  - Other required features? But we are very limited in manpower
- Other existing tool evaluation
  - from ALMA, Etienne?

I will share existing code and examples in the next days