



HAL
open science

Easy development of monitoring applications for an autonomous float

Sébastien Bonnieux, Mireille Blay-Fornarino, Sébastien Mosser, Yann Hello,
Guust Nolet

► **To cite this version:**

Sébastien Bonnieux, Mireille Blay-Fornarino, Sébastien Mosser, Yann Hello, Guust Nolet. Easy development of monitoring applications for an autonomous float. AGU Fall Meeting 2019, Dec 2019, San Francisco, CA, United States. . hal-03555745

HAL Id: hal-03555745

<https://hal.science/hal-03555745>

Submitted on 3 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Mermaid float

Objective:

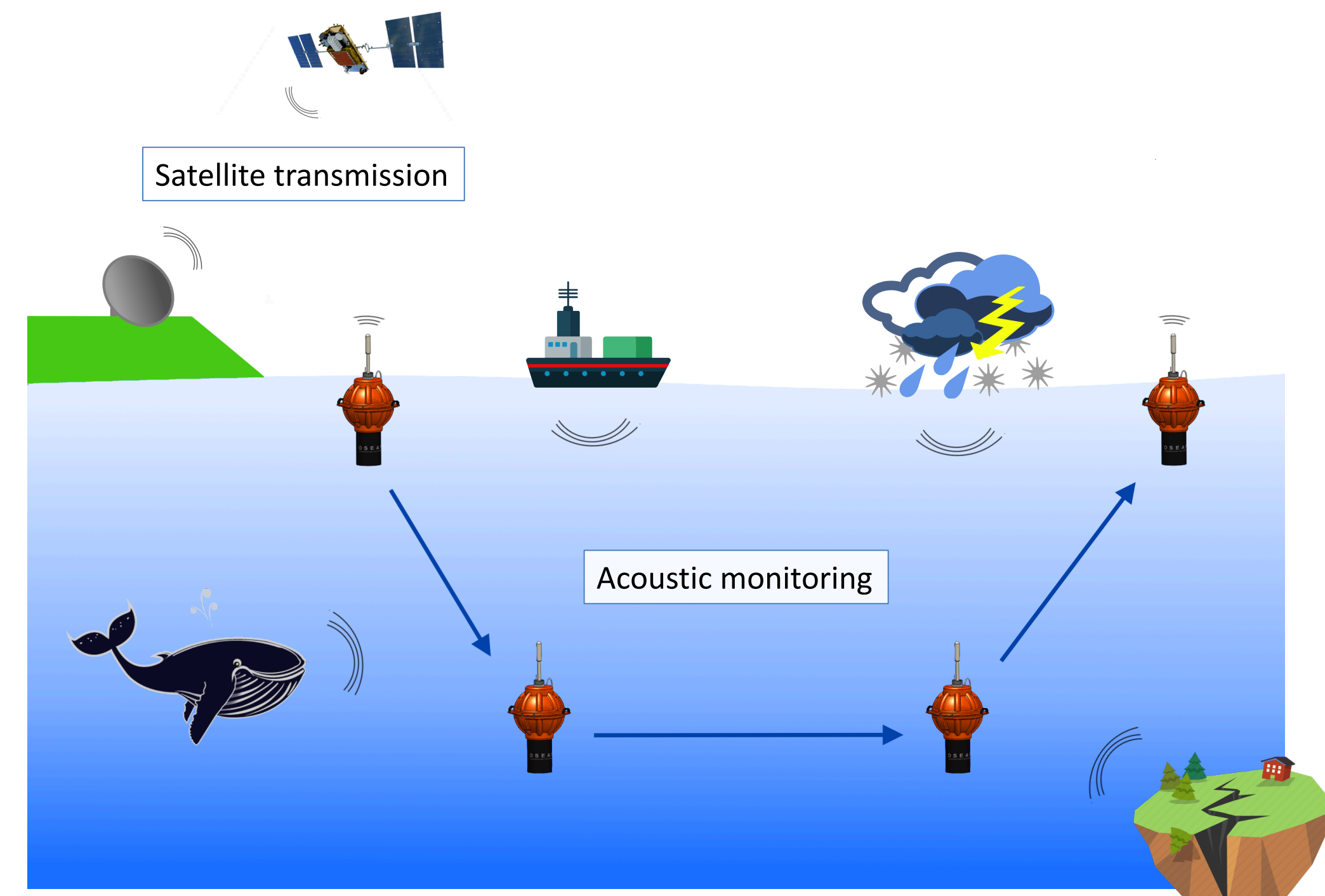
Enable global and multidisciplinary acoustic monitoring of the oceans with an adaptive autonomous float.

Problem:

Developing acoustic monitoring applications by traditional means is time requires much effort and expertise in embedded software.

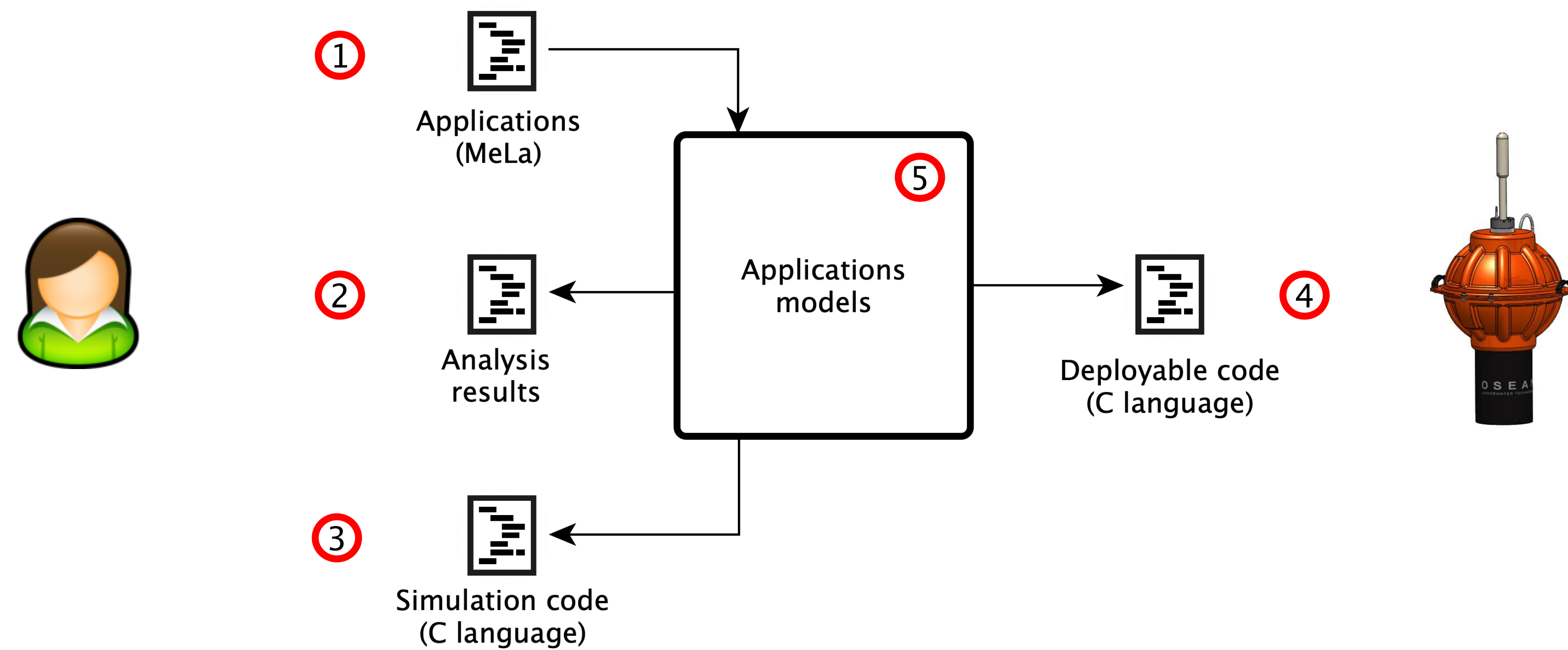
Proposition:

A Domain Specific Language (DSL) to allow biological, chemical, geological and physical oceanographers to develop applications by themselves.



Passive Acoustic Monitoring of the oceans has a lot of potential applications with whales, earthquake, rain or anthropogenic sound monitoring.

Model Driven Engineering approach



①	Scientists write applications with the Mermaid Language (MeLa), which hides embedded software concerns.
②	The analysis computes battery lifetime, satellite communication costs and processor usage of the applications.
③	Simulation of applications on a personal computer allows to verify the functional aspects of the applications.
④	The code for the instrument is automatically generated.
⑤	Applications transformations and analysis are done with models.

Syntax of the language

The mission configuration defines park time and depth of the float.

The coordinator defines when to execute acquisition modes.

Continuous acquisition mode.

Real time sequences have execution time constraints to guarantee the integrity of data.

Processing sequences allow to call function with long execution time.

Seismic application	Whales application
<pre> 1 # 1. Mission configuration 2 Mission: 3 ParkTime: 10 days; 4 ParkDepth: 1500 meters; 5 6 # 2. Coordination of acquisition modes 7 Coordinator: 8 ParkAcqModes: 9 Seismic; 10 11 # 3. Definition of a continuous acquisition mode 12 ContinuousAcqMode Seismic: 13 14 # 3.a. Input 15 Input: 16 sensor: Hydrophone; 17 data: x[40]; 18 19 # 3.b. Variables 20 Variables: 21 int[2400] lastminute; 22 bool detect; 23 float criterion; 24 transmitFile f; 25 26 # 3.c. Sequences of instructions 27 RealTimeSequence detection: 28 appendArray(lastminute, x); 29 detect = seisDetection(lastminute); 30 if detect: 31 @probability = 10 per week 32 call discriminate; 33 endif; 34 endseq; 35 36 ProcessingSequence discriminate: 37 criterion = seisDiscrimination(lastminute); 38 if criterion > 0.25: 39 @probability = 4 per week 40 recordIntArray(f, lastminute); 41 endif; 42 if criterion > 0.75: 43 @probability = 1 per week 44 ascent(); 45 endif; 46 endseq; 47 48 endmode; </pre>	<pre> 1 # 1. Mission configuration 2 Mission: 3 ParkTime: 20 days; 4 ParkDepth: 1000 meters; 5 6 # 2. Coordination of acquisition modes 7 Coordinator: 8 DescentAcqModes: 9 Whales every 10 minutes; 10 ParkAcqModes: 11 Whales every 3 hours; 12 AscentAcqModes: 13 Whales every 10 minutes; 14 15 # 3. Definition of a short acquisition mode 16 ShortAcqMode Whales: 17 18 # 3.a. Input 19 Input: 20 sensor: Hydrophone; 21 data: x[1024]; 22 23 # 3.b. Variables 24 Variables: 25 float presence; 26 int timestamp; 27 transmitFile f; 28 29 # 3.c. Sequences of instructions 30 ProcessingSequence identify: 31 presence = whalesDetection(x); 32 if presence > 0.2: 33 @probability = 1 per day 34 recordFloat(f, presence); 35 timestamp = getTimestamp(); 36 recordInt(f, timestamp); 37 endif; 38 endseq; 39 40 endmode; </pre>

Short acquisition mode.

The input defines the sensor to use and the variable containing the data.

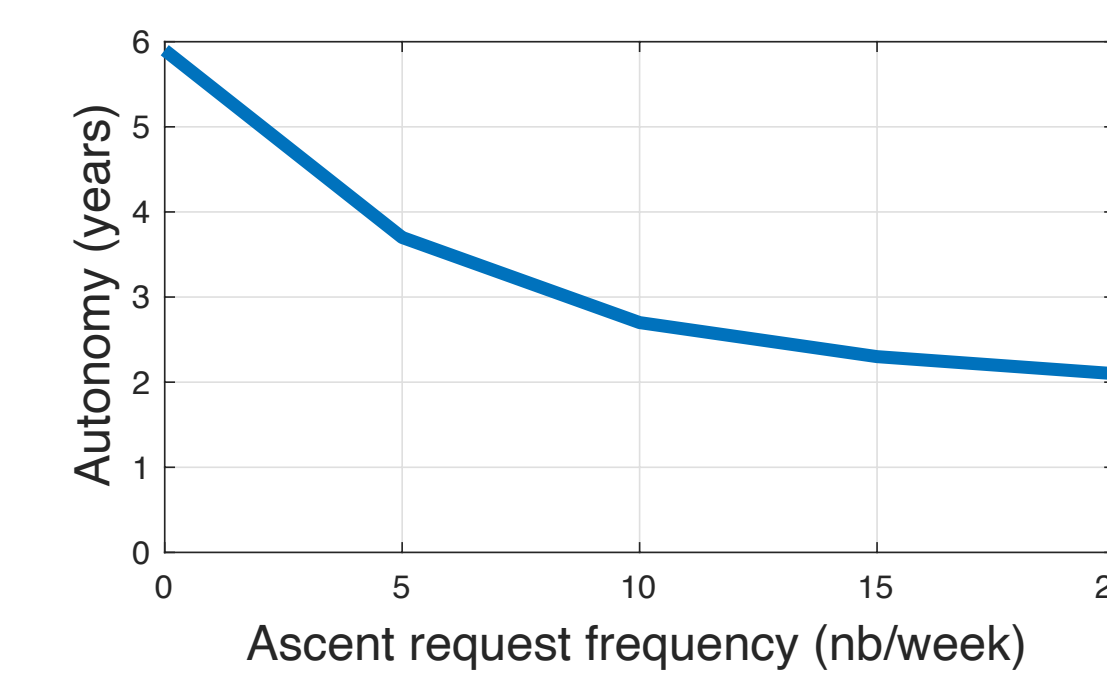
Conditions must be defined with a probability of execution used to estimate the battery lifetime and the amount of data transmitted through satellite.

Example of applications for earthquake and whales monitoring [1].

Developing applications with MeLa

Earthquake application

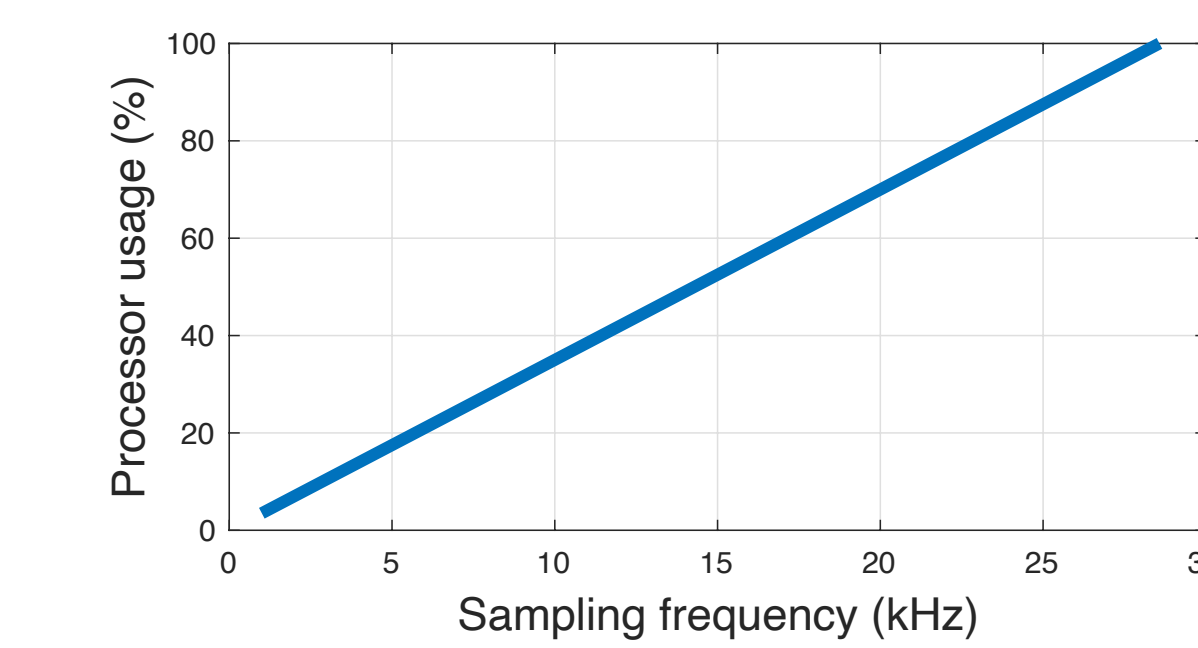
The algorithm consists of a STA/LTA algorithm that triggers a discrimination algorithm based on a wavelet transform [2].



The autonomy of the float is significantly reduced if too many earthquakes triggers the ascent of the float.

Marine mammals application

Computes the energy ratio of two frequency bands. A click is detected if the ratio exceeded a threshold. This algorithm is a simplified version of the ERMA detector [3].



Continuous acoustic monitoring is possible until a sampling frequency of 28 kHz.

Rain and drizzle application

Compare the acoustic signal level over two frequency bands. The spectrum is recorded for satellite transmission. This algorithm is a simplified version of the one described in [4].

Satellite transmission costs for different kind of records :

- Full spectrum: 350 \$
- Averaged spectrum: 40\$
- Event date: 1\$

Applications can be tuned during the development process to accord to the instrument limitations.

Conclusion

We have developed MeLa, a programming language dedicated to the development applications for an autonomous float. This language does not require to be an expert in embedded software and allows to consider the limitations of the instrument. We are working on a first pluridisciplinary version of the float, any new proposition of application is welcome.