Experiences and lessons learned developing a next-generation ground segment prototype

UNIVERSIDAD NACIONAL DE LA MATANZA
M.SC. ENG. PABLO SOLIGO – PHD. JORGE SALVADOR IERACHE

About us

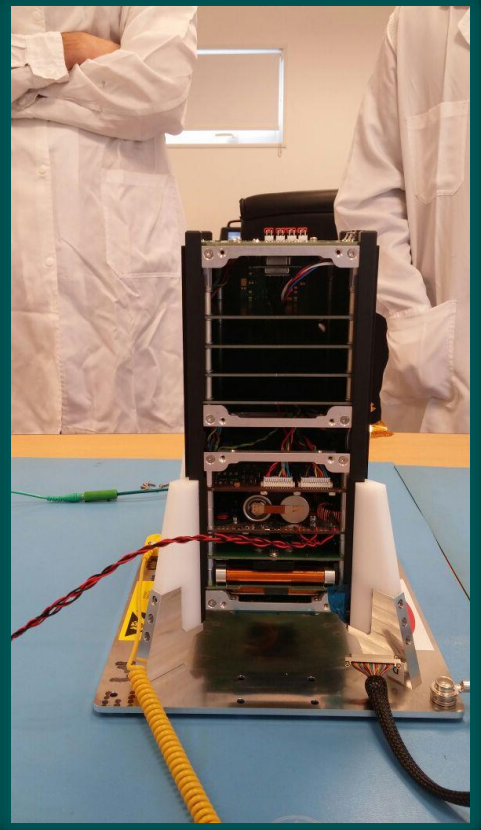
- M.Sc. Pablo Soligo
  - Teacher and researcher (UNLaM)
  - Software Engineering
  - 20 Years experience in software industry in different kind of projects
  - Space Master Course
- PhD. Jorge Salvador Ierache
  - Teacher and researcher (UBA/UNLaM)
  - M.Eng. Software Engineering (ITBA)
  - M.Sc. Knowledge Engineering (UPM)
  - Experience in development of aerospace surveillance and control systems

Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

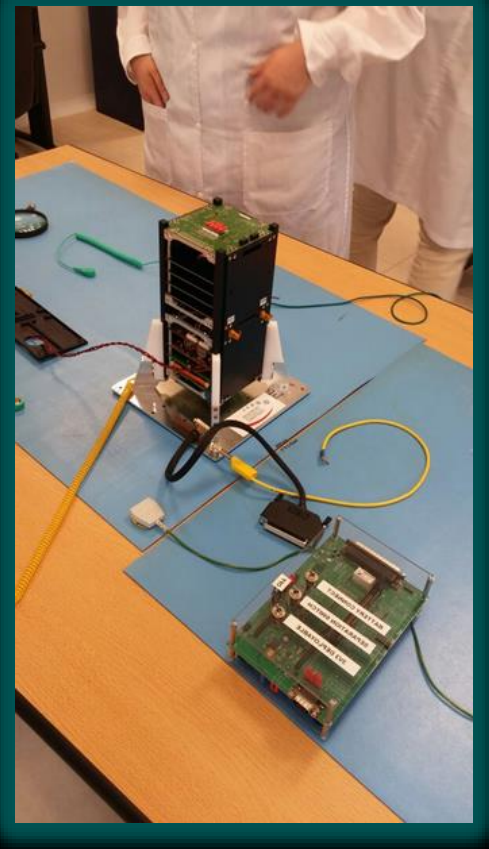
Schedule

- Early prototype.
- High level requirements and constraints.
- Space software solutions vs general purpose software solutions (Ground Segment).
- Our prototype today.
- Lessons learned.
- Future works.



Early Prototype

- MDIAE master course
  - Experience with ground segment systems from several agencies.
  - Satellite mission project
- Flight Segment: 2U cubesat.
- Reverse engineering



Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

High level requirements

- Multimission
- Multiplatform
- Cost-Effective
- Accessible for almost any platforms
- “Anywhere, anytime”

Working/Playing with space GS software (1)

Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

STUDIED SYSTEMS

- Persistence
  - Text Flat Files
  - Binary Files
- Communications and Distributed processing
  - Own Solution
  - CORBA

OUR PROTOTYPE

- Persistence
  - Relational Database
  - ORM
- Communication and Distributed processing
  - JSON
  - HTTP/HTTPS

"The database itself is a fixed format ASCII text file that can easily be read and modified by a standard text editor and no commercial database software (dBase, Oracle) is therefore required to maintain the database" Montenbruck, O., Eckstein, M. C., & Gonner, J. (1993).

"A second candidate for a database standard is one based upon a COTS relational database product such as Oracle."
Chaudhri, G. (2004). Ground Systems-The Need for Standardization. In *Space OPS 2004 Conference* (p. 178).

Working/Playing with space GS software (2)

Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

STUDIED SYSTEMS

- Command scripts and telemetry Calibration/Decoding
  - Own languages and interpreters (STOL/PLUTO)

OUR PROTOTYPE

- Command scripts and telemetry Calibration/Decoding
  - General purpose languages

*"Many Space-specific languages currently used for the development of operational procedures were defined decades ago and **are not used outside of the space industry**"* (Garcia, G. (2008, March) GSAW2008 Conference)

Languages comparison

	SPACE-SPECIFIC (eg. STOL)	GENERAL PURPOSE (eg. Python)
PROS	❑ (Sometimes) more user friendly for non-programmers ❑ Adapted to satellite operations ❑ High reliability	❑ Open source ❑ Very powerful ❑ Portable ❑ Language can be easily restricted / extended ❑ Wide availability of tools and programmers
CONS	❑ Proprietary language and/or tools ❑ Portability issues ❑ Limited, enhancements are expensive	❑ Potentially less readable if coding is not done carefully ❑ Too powerful?

Garcia, G. (2008, March). Use of Python as a satellite operations and testing automation language. In *GSAW2008 Conference, Redondo Beach, California*.

- Lack of
  - Documentation
  - Examples
  - Tools

General purpose language to command scripts and decode telemetry

Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

- We chose python because
  - Object oriented
  - Popular
  - Easy to learn
  - Powerful tools to develop and test
  - Examples and documentation
  - Multiplatform
  - **And Reflexive**

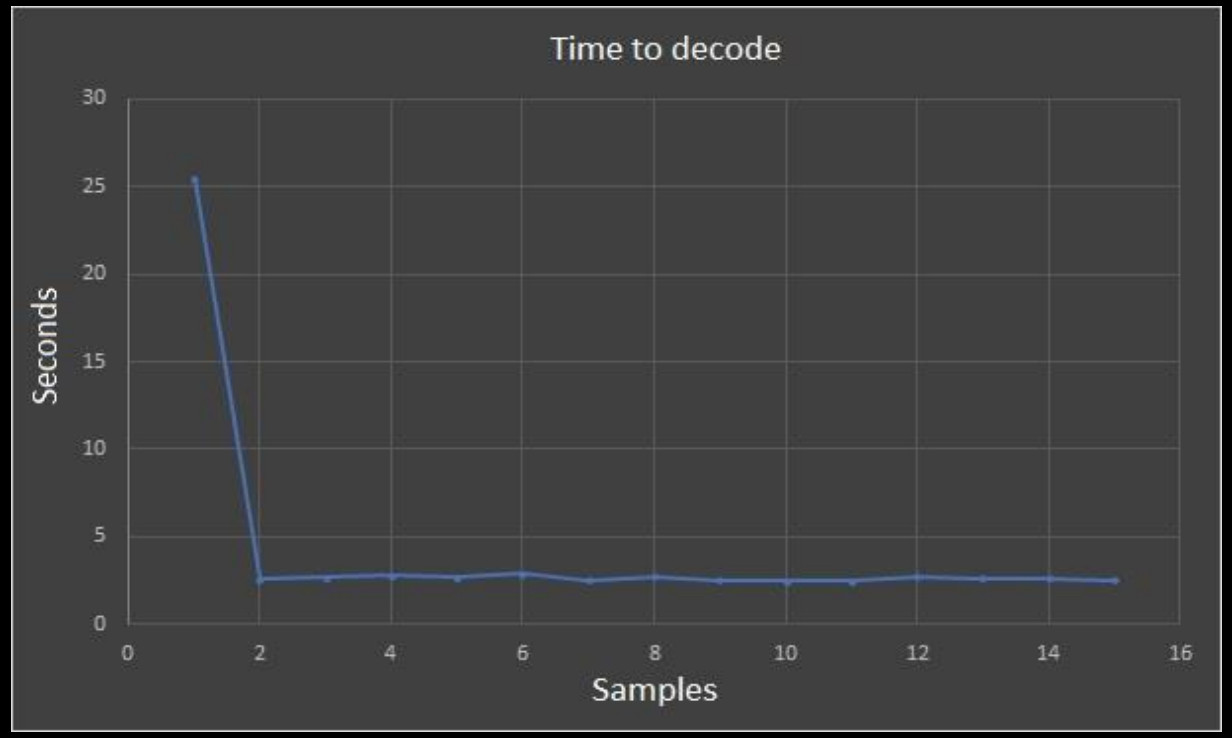
Rank	Change	Language	Share	Trend
1		Python	29.49 %	+4.5 %
2		Java	19.57 %	-2.4 %
3		Javascript	8.4 %	+0.1 %
4		C#	7.35 %	-0.4 %
5		PHP	6.34 %	-1.2 %

PYPL PopularitY of Programming Language(2019)

"Future support for proprietary languages and availability of tools is not guaranteed. Some operators have had serious problems replacing a system once the HW became obsolete." (Garcia, G. (2008, March) GSAW2008 Conference)

Reflection and the cost of processing

- Reflection is an expensive activity in terms of the cost of processing.
- Optimizations
  - Load the calibration functions and keep them in memory.
  - Reload only if there have been changes.
  - Apply the calibrations only if the raw value or function have changed.
- Applying these optimizations we managed to process 5000 raw values in 2.5 seconds (this amount of data is comparable with medium or big satellites) I.e. SAC-D/Aquarius.
- A packet every 8 seconds is an acceptable value for this kind of satellites.



The first sample takes 25 seconds because the calibration functions need to be loaded. Then 2.5 seconds is enough for complete packet calibration.

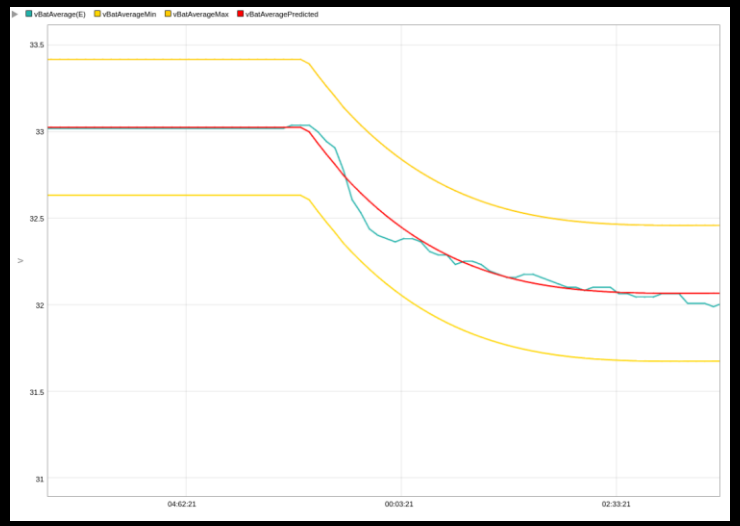
State of health

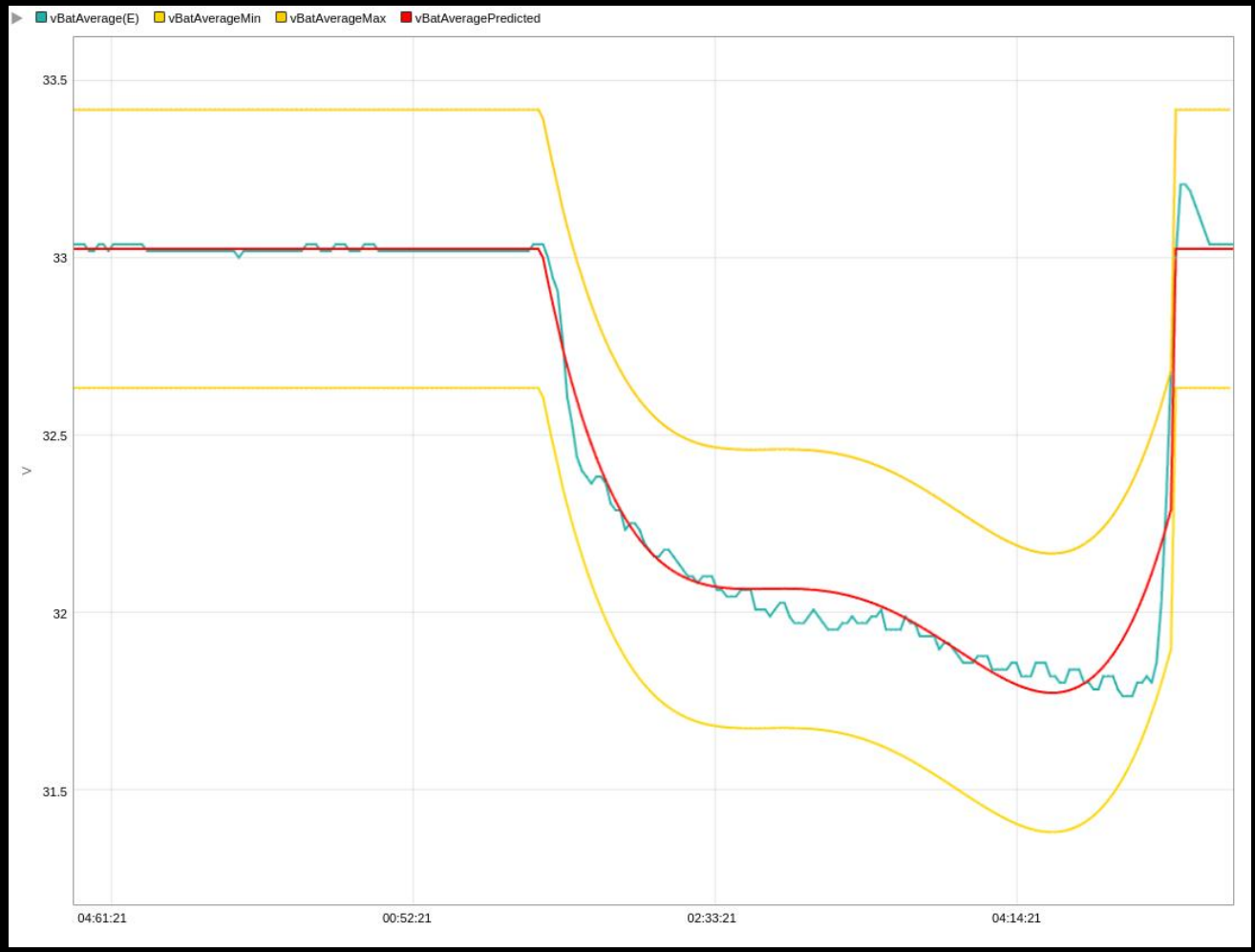
- We initially use limit check
  - Context insensitive
  - No value for history
- We use general purpose languages and interpreters to decode telemetry in runtime.
- We can harness the power of available libraries to predict the value of telemetry.
- We change limits dynamically based on predicted value.

Yairi, T., Nakatsugawa, M., Hori, K., Nakasuka, S., Machida, K., & Ishihama, N. (2004, October). Adaptive limit checking for spacecraft telemetry data using regression tree learning. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583) (Vol. 6, pp. 5130-5135). IEEE.
P. Soligo y J. S. Ierache, «Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos,» de XXV Congreso Argentino de Ciencias de la Computación (La Plata, 2019)., 2019.

State of health

- Dynamic limits changing with datamining and machine learning





Blue line: real value, red line: predicted value, yellow lines: dynamically changed limits.

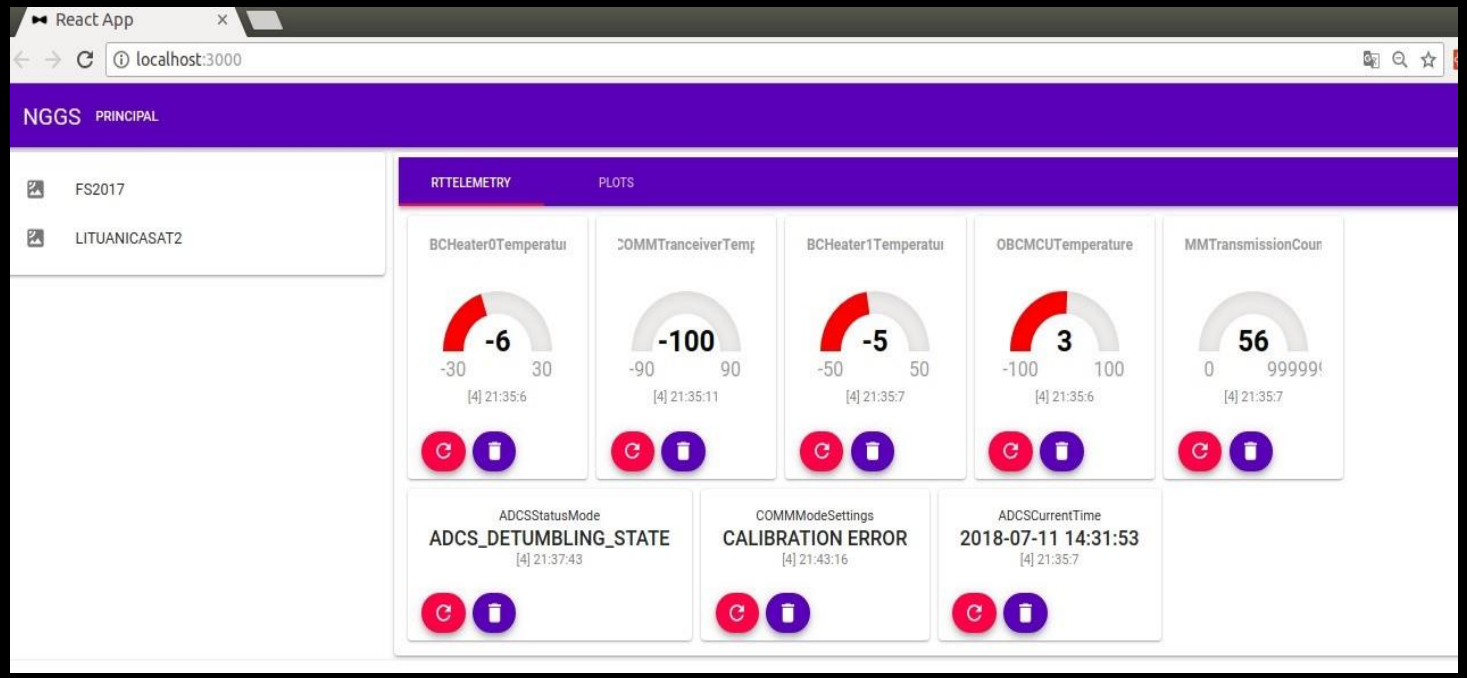
User interfaces

- Requirements
  - Accessible for almost any device
  - One development for any platform
  - Interactive
- Solution
  - SPA (Single page application)

First Prototype

We invested more than 120 hours achieving poor results, which only accounted for a fraction of the needs and had serious adaptability issues with different screen sizes.

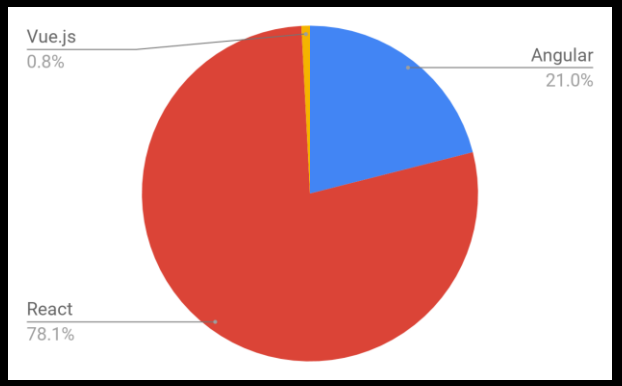
Finally we decided that we had to take another way...



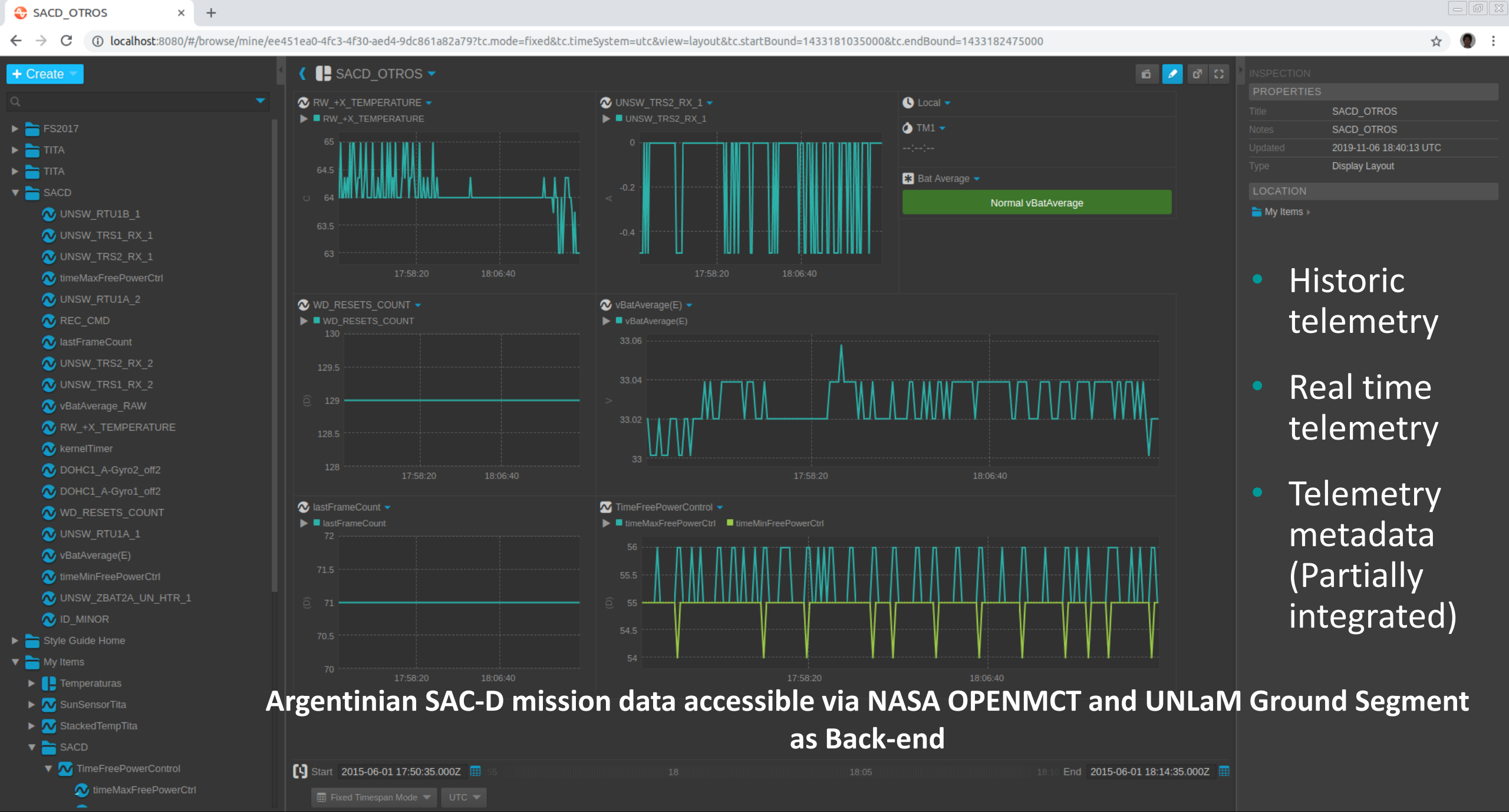
"Typically, the graphical user interface (GUI) of an interactive system represents about 48% of the source code, requires about 45% of the development time and 50% of the implementation time" (Myers and Rosson, 1992).

NASA OPEN MCT

- Open MCT is a next-generation mission operations data visualization framework. Web-based, for desktop and mobile.
- Open Source.
- Using Angularjs framework.
- Angularjs share market: 21.0%.



Indeed.com



- Historic telemetry
- Real time telemetry
- Telemetry metadata (Partially integrated)

Argentinian SAC-D mission data accessible via NASA OPENMCT and UNLaM Ground Segment as Back-end

Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

Lessons Learned and Results

TWO SIDES OF THE SAME COIN

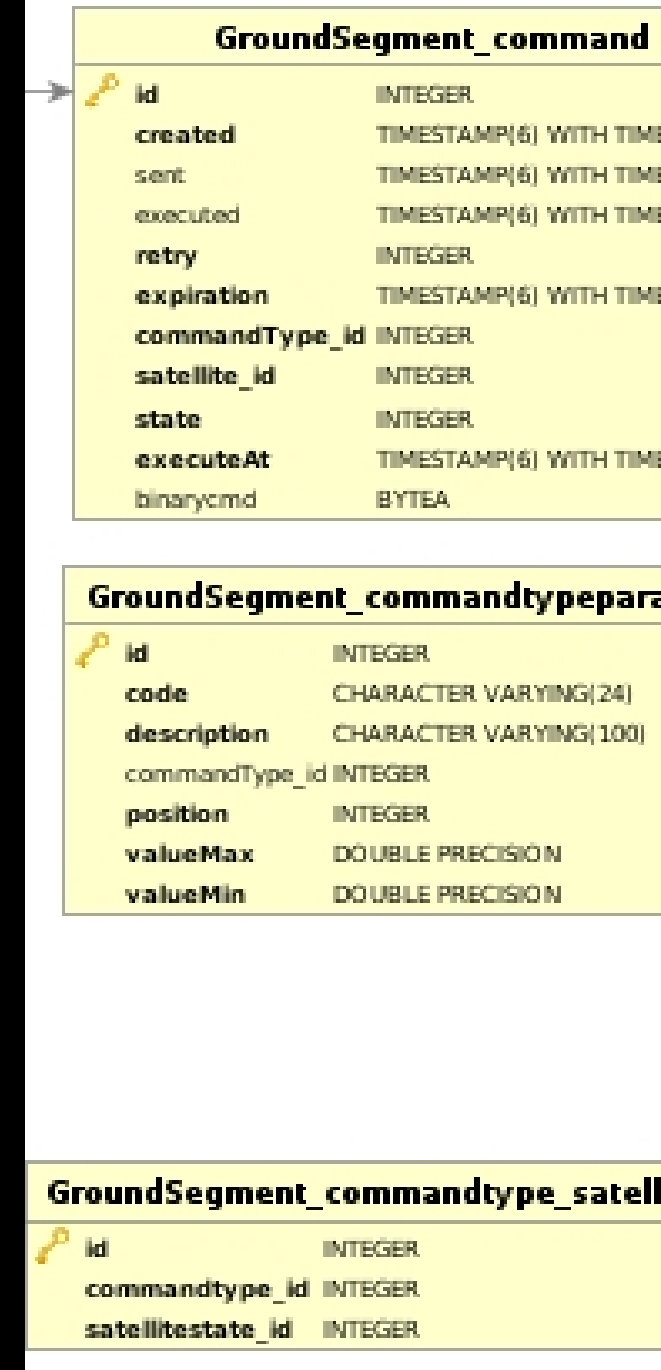
Lessons learned and results (1)

- Telemetry and telecommands
  - Telemetry
    - FS2017, Lituanicasat 2, Bugsat, SAC-D, And more*
  - Telecommands
    - FS2017, Simulators
- General purpose languages to command and calibrate/decode telemetry
  - Great and cheap option for small and even big missions.

*Data obtained from satnogs (https://satnogs.org/) and other resources.

Lessons learned and results (2)

- Telemetry mining and state of health
  - Promising low-cost alternative.
  - We must continue working on it.
- Relational Database/ORM
  - High productivity.
  - ORM is a mature tool now.



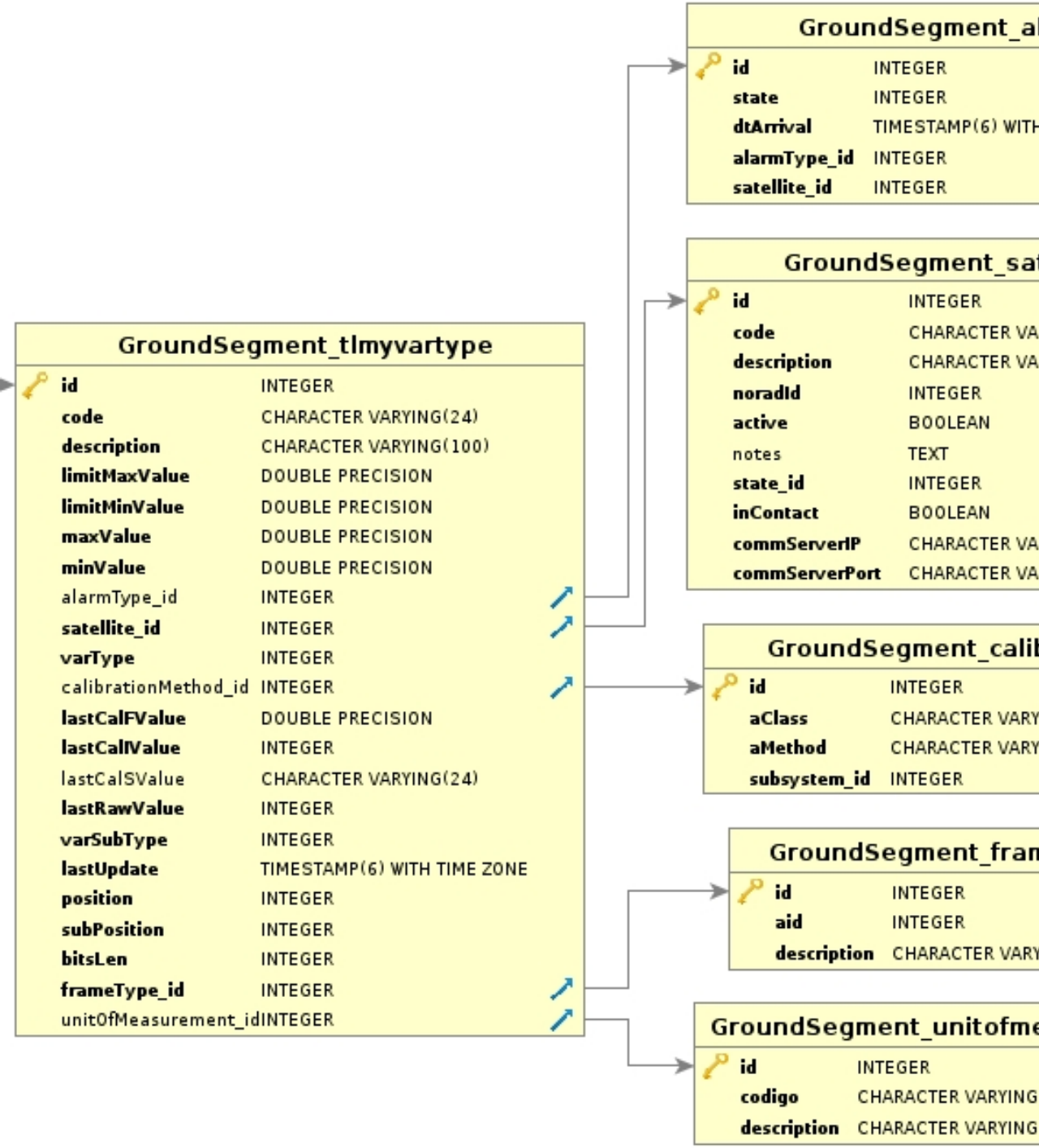
Lessons learned and results (3)

- User interfaces
  - Multidevice user interfaces could be a huge bottleneck.
  - SPA needs specialized or dedicated human resources.
  - NASA OPEN MCT, superior product with lack of examples.
  - NASA OPEN MCT needs specialized or dedicated resources too.
  - if you want to create a single application for any platform It is probable you will end up doing it with a single page app.

“Being a relatively new technology, a single page app requires great efforts and deep knowledge to troubleshoot…. Additionally, the code is hard to track, especially when you try to integrate the third-party elements like modules, plugins, and extensions” . *GEARHEART web development (2019, Jun 17). Pros and Cons of Building Single Page Applications in 2019 [Blog post]. Retrieved from https://gearheart.io/blog/pros-and-cons-building-single-page-applications-2019/*

Future works

- Amount of data
  - For high amount of data we should unnormalize de database design.
- State of health
  - Automatic search for correlated variables is still pending.



Future works

"Talks is cheap, show me code"

Linus Torvalds

- We are planning to publish the code Q3 next year.
- We must refactor some projects parts because It's a research-oriented prototype.
- There are errors design to correct.

Universidad Nacional de La Matanza

DIIT Departamento de Ingeniería e Investigaciones Tecnológicas

Thanks!

Questions?