PORÍS: practical-oriented representation for instrument systems

Jacinto Javier Vaz-Cedillo
Instituto de Astrofísica de Canarias – Vía Láctea s/n 38205 La Laguna, S/C de Tenerife (Spain)

ABSTRACT

This article presents a toolkit for defining simple but powerful **systems**. PORIS toolkit is an open and extensible source collaborative project that allows describing graph-based systems and their behavior in a snapshot. It provides a web editor for a domain visual specific language **(DSL**) and transformation tools to generate software **prototypes**, system configurations, specific **user interfaces** and documentation. Different kind of **instruments**, like the astronomical ones, can be described and represented using PORIS specifications and **models**. A significant advantage of using PORIS toolkit is that it makes easy and lighter providing instant feedback to domain experts in the dynamic process of defining new instruments.

Keywords: Systems, instruments, models, toolkit, sketch, prototypes, user interfaces, automatic software generation

1 INTRODUCTION

Nowadays the advanced astronomical instruments are engineering artifacts, so they are defined and built as highly automated systems electronically controlled. They are complex systems, so they are described using an architecture tree of subsystems collaborating together to get a specified behavior. To do a serious system engineering activity is a must, even in the early requirement caption stages.

PORIS is a software toolkit that can be used to make this activity easier, and it is very useful in the early specification stages when specifying the requirements of an astronomical instrument. Its use will give two important benefits to the developing team:

- The team will be able to 'sketch' the whole the instrument and its behavior in a single and understandable snapshot.
- As an added value, the team will receive, instantly, an automatically generated artifact very useful at that stage: a prototype of a configuration panel of the sketched instrument. The team will be able to interactively 'play' with this artifact in order to find some misunderstandings or errors in the concept they have of the instrument and its operating modes.

Analyzing the snapshot and 'playing' with the panel the team can, intuitively, synchronize the vision they share on the whole instrument. Obviously it is a macroscopic vision, a sketched vision, but at the early stages of instrument definition is worthy to have it synchronized and to have a good map of the instrument architecture and operating modes.

Figure 1. An instrument sketched in a single PORIS snapshot. It includes macroscopic architecture and behavior.

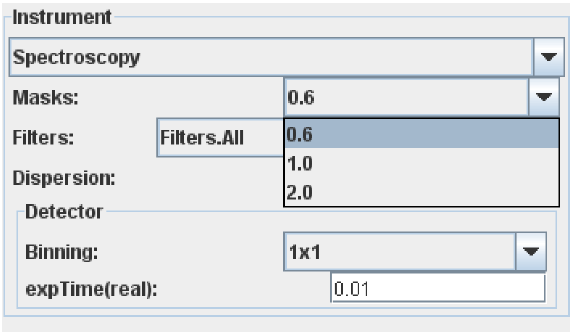


Figure 2. A panel that shows the interactive behavior of the instrument sketch.

The PORIS software toolkit is an open source project; it is available for free download and can be freely adapted to the needs of every team or instrument.

2 THIS ARTICLE'S ROADMAP

The author of this article strongly recommends the use of PORIS toolkit in the instrument development. To show the convenience of this toolkit, this article will follow these steps.

- The article will show arguments to demonstrate the convenience of using this toolkit.
- Then, an example of using PORIS toolkit helping the definition of an instrument will be shown, following these steps:
    - Brief introduction about the PORIS systems and visual notation used to describe them.
    - Construction of a PORIS sketch of a simple instrument, step by step.
        - Enumeration of the instrument and its subsystems.
        - Construction of the PORIS sketch architecture: instrument and subsystems.
        - Enumeration of the parameters and values of each instrument subsystem.
        - Addition of the parameters and values to the PORIS sketch.

        - Enumeration of the instrument modes, and description of how these modes impose restrictions to the parameters and values in the instrument subsystems.
        - Addition of the modes and restrictions to the PORIS sketch.
    - Snapshots of the instrument configuration panels, instantly generated by the PORIS toolkit from the PORIS sketch.
    - Demonstration of the 'agility' of the PORIS representation: quick response to specification changes.
        - Enunciation of a significant change in the instrument.
        - Modification of the PORIS sketch to respond to this change.
        - Snapshot of the configuration panels showing this change working.
    - A successful story of using PORIS toolkit: the Osiris instrument development project[1].
    - Finding more information and downloading the toolkit: PORIS toolkit project homepage.

3 PORIS SKETCH VS THE INSTRUMENT

3.1 Modeling vs. sketching

An instrument can not be completely defined using PORIS toolkit, because this is not a tool for modeling instruments. But it is very worthy to spend some time in "sketching" the instrument using PORIS toolkit.

Let us explain this using an analogy: a picture is not able to represent completely a person, but it is a good and effective representation of him/her in a lot of circumstances of the real life (as a recognition tool in the id card, as a way to remember past and happy times...)

Using PORIS toolkit we will obtain a fast, cheap, partial but useful representation of our instrument, so in this article we strongly recommend using it.

3.2 But, why not modeling?

The question is... why not directly using a tool capable of modeling the whole instrument and its behavior from the systems engineering point of view? The response is simple: the candidate tool has to meet some requirements and none of the systems considered did it. Our requirements were:

- Must be able to represent the whole instrument in a macroscopic way, showing the system-subsystem architecture.
- Must allow reusing the same subsystem in different systems.
- Must have a very simple notation and the user must not be required to know any language or standard to begin working.
- Must allow concurrent working, and the desired environment must be based on web pages, where possible.
- Must allow representing the behavior of instrument modes in a macroscopic, and must be able to represent the restrictions that a mode introduces on each instrument parameter.
- Must be extensible and, if possible, open source.
- When possible, maximize the use of standards.
- When possible, allow any team member to work on the instrument without using software developer tools.
- Must provide the complete representation of the instrument, subsystems, parameters, available values, modes and behavior in a single and understandable snapshot.
- Must provide some kind of instant interactive feedback of the instrument definition.

Some modeling alternatives have been taken into consideration but did not meet the requirements: Matlab/Simulink, UML tools, SysML tools, etc.

PORIS toolkit was built to meet these requirements, and offers it to the worldwide community as an open source project.

3.3 The convenience of using PORIS to sketch an instrument

PORIS is a toolkit to help the development; it does not 'solve' a need, so its use is not 'linked' to any need, but to convenience. This is very similar to the CAD tools: they have been not needed to design buildings during centuries, but nowadays all the architects use them to design every building.

This article has said that using PORIS toolkit is worthy. Let's see the concepts behind these words:

- It helps the team to refine the first specifications of an instrument, specially those visible from a macroscopic point of view:
    - The subsystems that compose the instrument, its parameters and the possible values of them.
    - The instrument observing modes, and how these impose restrictions on the subsystems, parameters and values that compose the instrument.
- The team can use PORIS to automatically generate some added values to be used during the instrument development and life, as instrument configuration panels, documentation, etc.

3.4 Differences between the PORIS sketch and the instrument model

Astronomical instruments are similar to PORIS systems in some aspects:

- Both are mainly based in tree architectures: systems that contain subsystems that can contain other subsystems avoiding loops in this definition.
- Both have behaviors mainly defined by the modes they can adopt.
- Their subsystems may have parameters that, in most cases, take simple values. For instance, a filter loader takes as value a filter from a list of filters available, so this is taking a filter id from an enumeration of filters ids. The exposure time of the camera subsystem takes a value in seconds between 0 and a maximum number, etc.
- Their modes are mainly defined by the restrictions they impose to the parameters of the subsystems.

But a PORIS system differs a lot from an instrument in some other aspects:

- An instrument has states, like a finite state machine, a PORIS system does not.
- An instrument executes sequences, a PORIS system does not.
- An instrument manages complex information (as the acquired images), the PORIS system does not.
- The change of a parameter value in a PORIS system is 'free': does not consume time or resources, but in an instrument, frequently, this is not possible. Normally, the change of the value of a parameter includes a configuration sequence that executes this change step by step. For instance, a filter must be unloaded and stored in the filter storage, and the new filter must be loaded from the filter storage.
- The change of a mode in a PORIS system is 'free': does not consume time or resources, but in an instrument, frequently, this is not possible. Normally, the change of the mode of an instrument implies a sequence of re-configuration of the instrument, activating and deactivating subsystems and changing parameter values, so implies a sequence that contain another sequences, wasting time and resources.
- In a PORIS system, the behavior of a subsystem that has been disabled is similar to the no-existence of it. In fact, the subsystem does not exist from other subsystems point of view. Only the memory of the subsystem configuration remains, to allow the parameter values to persist until next activation of the system. In an instrument, the deactivation of a subsystem implies a sequence of deactivation that puts the subsystem in a secure state that tries to minimize the interferences of this subsystem on the other parts of the instrument.

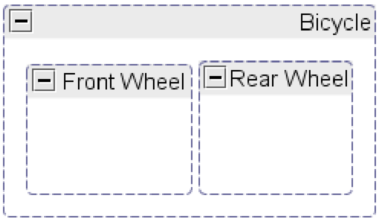
While sketching the instrument, the similarities must be taken into account and the differences must be avoided.

4 A BRIEF INTRODUCTION TO PORIS SYSTEMS

The language used by PORIS toolkit - from now 'PORIS language' - can represent very simple systems. This language can be used in a graphical way. In this section we introduce the PORIS language graphical notation, in order to use this notation to illustrate the example to be developed in the next section.

4.1 PORIS notation: system architecture description.

A PORIS system is composed of subsystems.



Notation:

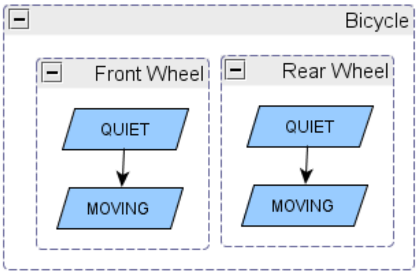
A system is a box. Inside the system we can find subsystems, which are represented as boxes too. In our notation, the only difference between system and subsystem is that the system is not included inside another subsystem.

Description:

A Bicycle is composed by a Front Wheel and a Rear Wheel.

Figure 3. Bicycle architecture in PORIS notation.

Each subsystem may have parameters that can take values.



Notation:

Values are represented as parallelograms. Some parameters can organize their possible values in enumerations, showing the natural order of the choices. This is represented by arrows that denote this natural order.

Description:

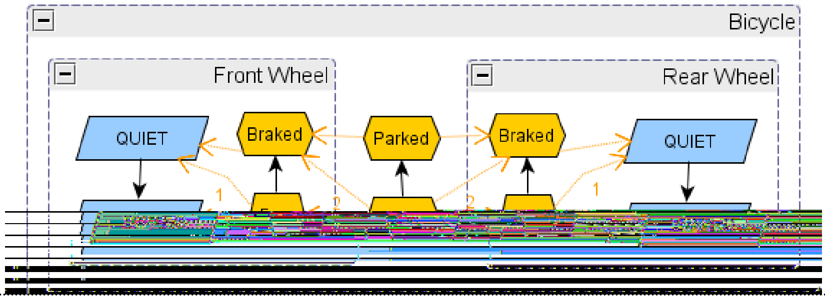
Both wheels may take the value QUIET o MOVING.

Figure 4. Bicycle architecture in PORIS notation with values.

4.2 PORIS notation: system behavior description.

In PORIS notation, the behavior of a system is defined by the restrictions the modes impose on the parameter values. These restrictions are modeled as a model cascade. Each subsystem offers to their client subsystems a collection of possible modes.

A mode is defined as two subsets: the possible values of the parameters of the subsystem, and the possible modes of the other subsystems used by the present subsystem.



Notation:

Modes are drawn as hexagon-shaped boxes. Values activated by them are denoted by arrows beginning in the mode and ending in the possible values. Sub modes are also denoted the same way.
If needed, a number is used to indicate the precedence order between the selected values or modes, if different from natural order.

Figure 5. Bicycle architecture and its (strange) behavior in PORIS notation.

Selecting a mode in the highest level, the system will propagate this selection to lower subsystem modes, so every parameter in every subsystem present in the system can be affected by that change. This solution has some interesting advantages:

- No "server" subsystem needs to know any detail of the "client" subsystems that use it.
- No "client" subsystem needs to know the internship of the "server" subsystems (parameters, values, subsystems that compose them). It only needs to know the set of modes that each "server" subsystem offers to their clients.

Team can reuse a subsystem in different subsystems, or to substitute easily a subsystem by another one without a complex interface to rebuild. This also forces each subsystem to define its responsibilities, behavior, etc. in the semantics of its own scope.

5 USING PORIS TO DEFINE INSTRUMENT SKETCHES

Let's define a simple instrument and then let's sketch it in PORIS language.

5.1 Sketching the instrument architecture

The technical concept under the sample instrument is very simple: the light will pass through the following devices: a mask, a filter, a dispersion element (in our sample instrument the devices used are grisms), finally, will be captured by the detector. Some of these stages are optional depending on the observing mode used. There are multiple masks, filters and grisms available to be used by the instrument, so we can say that each stage is managed by a subsystem that controls the loading and unloading of the devices in the light way. With this information the first architecture of the instrument can be sketched:

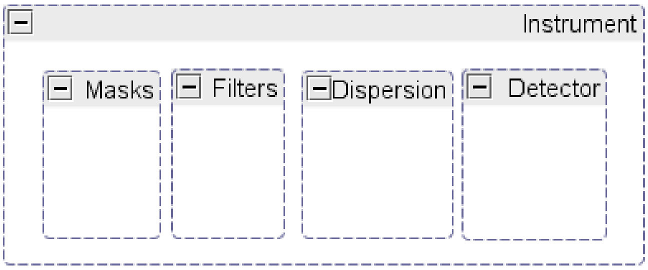


Figure 6. PORIS sketch, the instrument architecture.

Let's define the parameters and values of the instrument.

- The available masks are:
    - A 0.6 arc sec long slit.
    - A 1.0 arc sec long slit.
    - A 2.0 arc sec long slit.
- The available filters are:
    - Red
    - Blue
- The grisms are:
    - R500
    - R1000
    - R2000
- The detector parameters are:
    - Exposure Time: the range of possible values is from 0 to 1000 seconds, 0.01 as default value.
    - Binning:
        - 1x1
        - 2x1
        - 1x2
        - 2x2

[redacted]

Notation:

To represent the valid range of values of a continuous parameter, the minimum, default and maximum values are represented as circles inside a parallelogram.

Figure 7. PORIS sketch: parameters and values.

5.2 Sketching the instrument behavior

This is the expected behavior of the sample instrument to be sketched:

- The instrument will have two operating modes:
  - Photometry:
    - No mask allowed.
    - No grism allowed.
  - Spectroscopy:
    - A mask must be present.
    - A grism must be used.
    - The only two detector binning allowed are 1x1 y 2x2.

After adding these behaviors to the PORIS sketch:

Figure 8. PORIS sketch: modes.

The changes in the sketch are:

- At Instrument level, the Spectroscopy and Photometry modes were added.
- At Masks, Dispersion, Filters and Binning subsystems, the mode All was added. This model enable all the possible values of that subsystem parameter.
- At expTime subsystem, the mode Normal has been added, to enable the expTime value range.
- Photometry does not use Dispersion, so there is no arrow between Photometry and any Dispersion mode.
- Photometry does not use Masks, so there is no arrow between Photometry and any Masks mode.
- Both Photometry and Spectroscopy can use a filter or not, so we have added a mode call None in Filters, that does not activate any filter.
- Spectroscopy will need a reduced set of Binning values (only 1x1 and 2x2), so a mode called Square has been added to Binning parameter.
- Photometry and Spectroscopy must be able to change values in Binning and expTime, so some Detector modes have been added, in order to link the Instrument modes with the Binning and expTime modes. The Detector's Normal mode will activate the mode Normal in expTime and the mode All in Binning, while Detector's Square Binning mode will activate the mode Normal in expTime and the mode Square in Binning.

5.3 Configuration panels generated from PORIS sketch.

Once a PORIS sketch of an instrument is drawn, the PORIS toolkit can automatically generate some interactive configuration panel. In a few minutes, the team can be 'playing' with an interactive artifact and can check the mode restrictions effects.

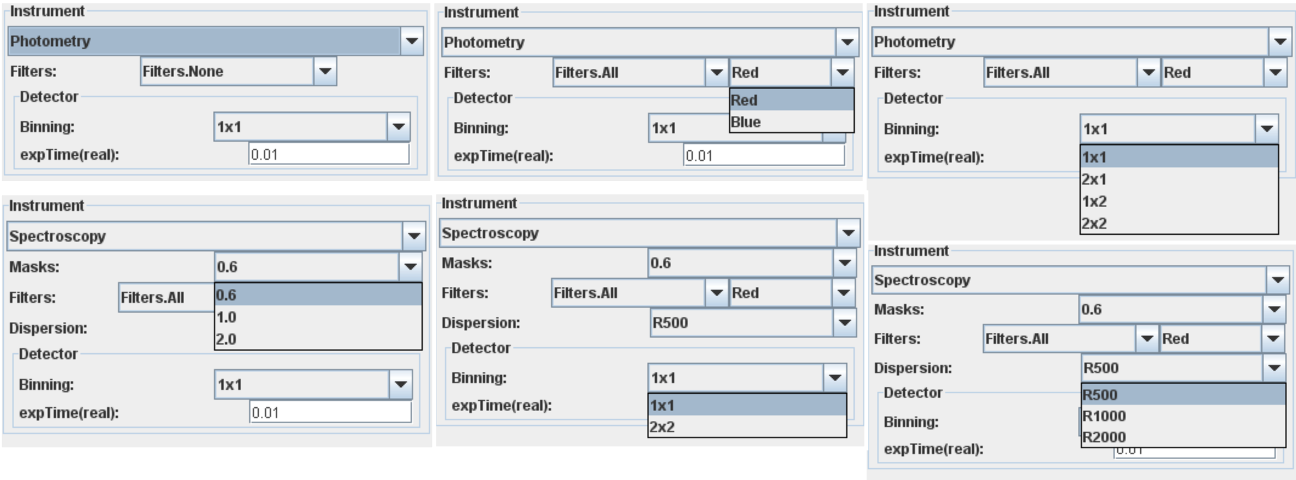


Figure 9. Six screenshots of the configuration panel generated by PORIS toolkit.

As can be seen in figure number 9, a change in a mode selector (photometry to spectroscopy or ‘Filters.none’ to ‘Filters.all’) causes the rest of the panel to be recalculated under the restrictions imposed by the new mode.

5.4 An example of quick response to changes (Agility).

In the last years, engineering processes have accepted the agility as a key concept in the maximization of their success probabilities. They have assumed that the development context is always changing, and this includes the specification. Since these changes are expected, the processes and tools must be sensitive to them and giving a fast response to this changing environment is a must requirement for them.

PORIS is able to give a fast response to change in the specifications, since a PORIS sketch is very easy to change and the automatically generated panels can instantly show these changes interactively to the team.

Let us plan an example of change in the specification of our sample instrument and show how this change is introduced in the PORIS sketch:

- When choosing for detector's CCD candidates, some of them presented a “faster mode” that allows to use half of the detector’s field of view to capture the image while using the other to store the previous one. This provides the ability to get images faster at same resolution, if the observation to be done needs less than half detector..
- The team agrees that there is a valid science case for this new functionality. They decide to create a new instrument observing mode to offer to the client scientists, called ‘Faster Photometry’.
- A special mask must be used to hide the other half of the CCD. The new mask is called ‘Hald Field’ and must be accessible only to the Faster Photometry mode.

These steps are followed to add this new change to the PORIS sketch:

- A new value called Half Field' is added to the Masks subsystem.
- A new mode called 'Faster Photometry' is added to the instrument. This activates the following sub-modes:
    - In Masks, Half Field mode is enabled.
    - In Filters, None and All are enabled.
    - In Detector, Faster mode is enabled.
    - In Dispersion, no mode is enabled.

Figure 10. Modifying the PORIS sketch.

The team instantly obtains the configuration panels changed.

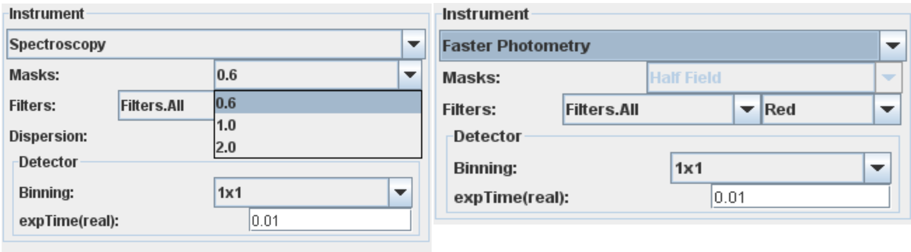


Figure 11. Configuration panels give instant feedback after changes addition.

6 A SUCCESSFUL USE OF PORIS TOOLKIT: OSIRIS AT GTC

PORIS systems have been successfully used in the Osiris[1] project. These are examples of tasks that have used the PORIS toolkit. The Osiris project is far older than PORIS toolkit, so it has not been used to refine the instrument specifications, but there are a lot of tasks that depend on the subsystems and modes of the instrument, and then a PORIS sketch was made to obtain the benefits given by the tool.

- Used to capture the requirements for the Osiris[3] configuration panels: PORIS toolkit was used to give to the science team configuration panels from the user point of view. The science team interactively "played" with them and then gave feedback to redefine the instrument configuration software. In fact, PORIS toolkit has been developed in response to the difficulty of capturing the requirements from the software engineer point of view.
- When developing the Osiris Mask Designer software tool, the observation configuration panel for MOS and the software preferences panel were automatically generated using PORIS toolkit.

6.1 PORIS sketch of Osiris Instrument

These are some snapshots of the Osiris instrument PORIS sketch:

Figure 12. PORIS sketch of the whole OSIRIS instrument.

Figure 13. PORIS sketch of the OSIRIS filters subsystem.

6.2 PORIS Configuration panels and their integration in software tools.

Configuration panels have been successfully integrated into Osiris software tools.

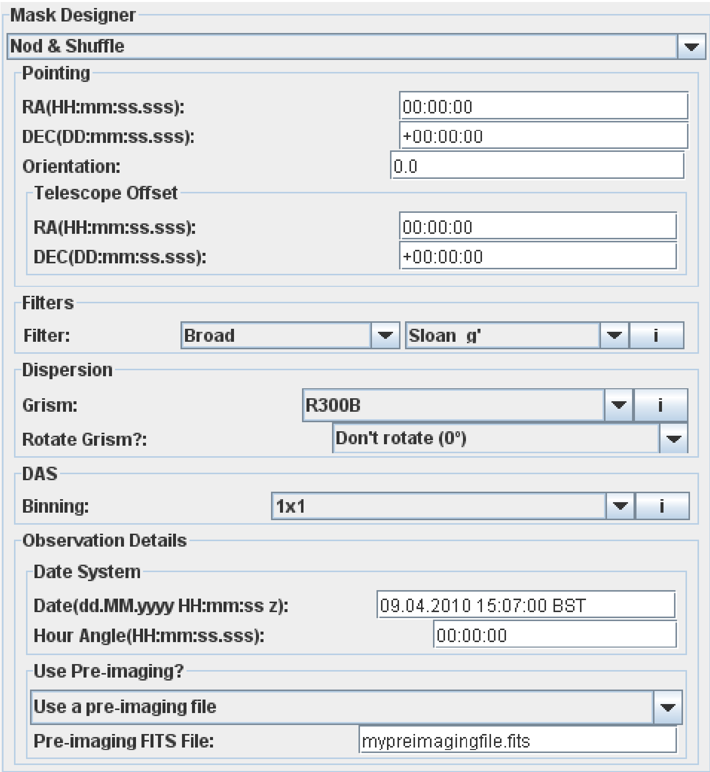


Figure 14. OSIRIS configuration panel integrated in Osiris Mask Designer software tool.

The Osiris Mask Designer searches for instrument xml file and, in execution time, show the panels to the user. Using this method, if the instrument changes its architecture or behavior in the future (for instance adding some filters, grisms or modes), the software does not need to be changed, because the only change needed is to update the instrument xml file. The instrument xml is a PORIS sketch of the instrument exported to an XML file.

7 FINDING INFORMATION ABOUT THE PORIS PROJECT

PORIS toolkit can be freely downloaded from its project homepage at http://www.elporis.com.

More documentation and contact information on the PORIS toolkit software can be also found at the project's home page.

REFERENCES

[1] Osiris project information can be found at http://www.gtc.iac.es/en/pages/instrumentation/osiris.php
[2] PORIS toolkit project homepage and information can be found at http://www.elporis.com
[3] Cepa, J. et al., "OSIRIS tunable imager and spectrograph", Proc. SPIE 4008, 623 (2000).