



**PORGY : a Visual Analytics Platform for System  
Modelling and Analysis Based on Graph Rewriting**  
Bruno Pinaud, Oana Andrei, Maribel Fernández, Hélène Kirchner, Guy  
Melançon, Jason Vallet

► **To cite this version:**

Bruno Pinaud, Oana Andrei, Maribel Fernández, Hélène Kirchner, Guy Melançon, et al.. PORGY : a Visual Analytics Platform for System Modelling and Analysis Based on Graph Rewriting. 2017. hal-01461513

**HAL Id: hal-01461513**

**<https://hal.archives-ouvertes.fr/hal-01461513>**

Submitted on 8 Feb 2017

**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.

# PORGY : a Visual Analytics Platform for System Modelling and Analysis Based on Graph Rewriting

Bruno Pinaud\*, Oana Andrei\*\*, Maribel Fernández\*\*\*  
Hélène Kirchner\*\*\*\*, Guy Melançon\*, Jason Vallet\*

\*Université de Bordeaux, LaBRI, France, {prénom.nom}@u-bordeaux.fr

\*\*School of Computing Science, University of Glasgow, UK

\*\*\*King's College London, UK

\*\*\*\*Inria, France, helene.kircher@inria.fr

We propose PORGY<sup>1</sup> a visual modelling framework (Fig. 1) designed for specifying, simulating, and analysing complex systems. PORGY is built on top of the open-source visualisation framework TULIP<sup>2</sup>. PORGY is based on the use of *port graphs with attributes* to represent system states. In a port graph, edges connect to nodes at specific points, called ports. Nodes, ports and edges describe the system components and their relationships, while attributes encapsulate the data values associated with entity. We use graph transformations based on port graph rewrite rules to describe the evolution of the system.

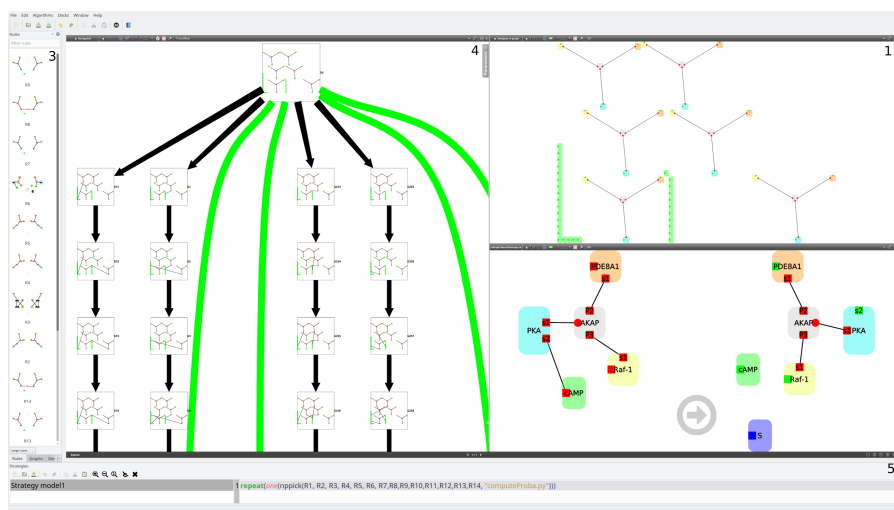


FIG. 1 – Overview of PORGY: (1) editing a graph; (2) editing a rule; (3) all available rules; (4) the derivation tree, a complete trace of the computing history; (5) editing a strategy.

1. <http://porgy.labri.fr>

2. <http://tulip.labri.fr>

Port graph rewrite rules are graphical representations of transformations in the system, thus they provide a direct, visual mechanism to observe the system's behaviour. In addition to port graphs and rewrite rules our modelling approach includes *strategy expressions* to steer rule applications. Strategies allow to use operators to combine graph rewriting rules, as well as operators to define the location where rules should, or should not, apply. Often more than one transformation is possible at a given state, in which case instead of a single transformation step, we may have several alternatives to choose from, and in turn, generate several different sequences of transformations. The various transformation sequences are organised as a tree structure, which we call *Derivation Tree* (DT).

In order to support the tasks involved in the study of a graph rewriting system, PORGY provides facilities to view each component at the same time (rules, strategy, any state of the rewritten graph, DT), to perform on-demand rewriting (strategy-based or rule-based) with drag-and-drop mechanisms, to synchronise the different views to track evolution of system properties, to explore a DT with all possible derivations at different scales, to track the rewriting process throughout the whole DT, or to plot the evolution of a chosen parameter.

Generally speaking, PORGY has been designed with the Visual information-seeking mantra of Shneiderman (1996) in mind: *Overview first, zoom and filter, then details on demand*.

*Overview First.* To understand the behaviour of non-deterministic systems, it is often useful to execute several times the same rewrite program on the same input to look for potential variations. These can be seen as branches in the DT. Although it is often a large data structure, it provides an indexed representation of the system evolution where each node represents one system state, and an edge is the application of a rule or a strategy. PORGY allows to analyse the derivation tree and work with it at different levels. For instance, Small Multiples (SMs) allow to see consecutive graph states like a comic-strip.

*Zoom and Filter.* One may be interested in plotting the evolution of a parameter computed out of each intermediate state. An interactive scatter plot can be built. Moreover, thanks to the TULIP backend, all graphical views are synchronised. For instance, if some interesting points are selected inside the scatter plot, they are also immediately selected inside the corresponding branch of the derivation tree. The synchronisation is also valid for graph elements.

*Details on Demand.* We can investigate further the selected nodes by zooming in and seeing distinctly the graphs. Hovering the mouse pointer over an edge allows to see which elements were changed by the application of the rule. The modified elements are emphasised in the picture, to clearly display which ones have evolved.

In this short abstract, we have introduced some key features of PORGY, an open-source general-purpose modelling and analysis environment based on graph rewriting. Domain-specific versions of PORGY can be easily implemented by extending or refining the features presented here thanks to the TULIP plugin system.

## References

- Fernández, M., H. Kirchner, and B. Pinaud (2016). Strategic Port Graph Rewriting: An Interactive Modelling and Analysis Framework. Research Report, Inria.
- Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symp. on Visual Languages*, pp. 336–343.