# Presheaves for Processes and Unfoldings *

**Clovis Eberhart[1] and Tom Hirschowitz[2]**

**1 Université Savoie Mont-Blanc**
**2 CNRS, Université Savoie Mont-Blanc**

Baldan et al. [1] give a general framework for rewriting systems based on adhesive categories. They give a definition of *process* in their setting, which is a sequence of rewriting steps considered up to permutation of independent steps, and they build the *unfolding* of any object. Here, we propose a different approach to model rewriting, processes, and unfoldings, which is based on a formalism inherited from *playgrounds* [3]. The main contribution of our work is a direct, combinatorial description of processes and unfoldings in terms of presheaves. Just like in Bladan et al.'s work, the unfolding of an object has a universal property (slightly different from theirs, though very close in spirit). This framework can adequately model executions of Petri nets, term graphs, or interaction nets, as well as processes in simple concurrent languages such as CCS [3], the $\pi$-calculus, or the join-calculus. It can also model some form of graph rewriting, which has yet to be compared to existing frameworks, such as SPO and DPO. We are submitting this to CALCO Early Ideas with the hope that the community will give us some feedback, mainly on the kind of questions that this framework could help solve.

The main difficulty in Baldan et al.'s work is that a process is more than the sequence of its successive positions: it also records all the different occurrences of the rules that were applied. In our framework, this is dealt with by describing the whole process as a single presheaf that describes a combinatorial object of higher dimension. Given a signature $\Sigma$ that describes a set of objects (which are the presheaves over a category $\mathbb{C}$) and rewriting rules on these objects (given as a set $\mathcal{R}$ of spans of such objects), we build a bicategory $\mathbb{B}$ that models rewriting on this set of objects according to the given rewriting rules. This bicategory will have as objects the set of objects described by $\Sigma$, and as morphisms from an object $Y$ to an object $X$ all the processes (also called *execution traces* or *rewriting traces*) that correspond to rewriting $X$ into $Y$ according to the rules given by $\Sigma$. Processes are defined as particular cospans in $\widehat{\mathbb{C}[\mathcal{R}]}$ (the cateogry of presheaves over some category $\mathbb{C}[\mathcal{R}]$ constructed from $\mathbb{C}$ and $\mathcal{R}$). Such a cospan $Y \to U \leftarrow X$ describes an execution trace whose starting position is $X$ and whose final position is $Y$. We build $\mathbb{C}[\mathcal{R}]$ from $\mathbb{C}$ by adding an object $r$ for each rewriting rule in $\mathcal{R}$ and some meaningful morphisms from objects of $\mathbb{C}$ to $r$. We then define *seeds*, which are the local shape of a rewriting step, by equipping each representable presheaf $\mathsf{y}_r$ with its initial and final positions $X$ and $Y$ (as given by the rewriting rule), thus forming a cospan $Y \to \mathsf{y}_r \leftarrow X$. Rewriting steps are then defined by embedding seeds into larger positions (thus allowing rewriting steps to occur in context). Finally, rewriting traces are defined to be compositions of rewriting steps in the bicategory of cospans in $\widehat{\mathbb{C}[\mathcal{R}]}$. We then derive the unfolding $U_X$ of any object $X$ as the colimit of all the rewriting traces whose starting position is $X$.

Under some assumptions on the category $\mathbb{C}$, the presheaves over $\mathbb{C}[\mathcal{R}]$ have a graphical notation that is both close in spirit to string diagrams, and also inspired by that of graphs (but with multiple types of edges and higer-dimensional edges). For this reason, we usually call them *higher-dimensional* graphs or string diagrams (though none of these terms are actually very accurate).

* The authors acknowledge support from French ANR projets blanc Récré ANR-11-BS02-0010.

6th Conference on Algebra and Coalgebra in Computer Science.
Editors: Larry Moss and Pawel Sobociński; pp. 1–2

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The interested reader can find an online draft [2] in which the machinery sketched here is explained in detail, together with the running example of a Petri net to illustrate the notions of seed, rewriting step, process, and unfolding.

## References


1 Paolo Baldan, Andrea Corradini, Tobias Heindel, Barbara König, and Paweł Sobociński. Processes and unfoldings: concurrent computations in adhesive categories. *Mathematical Structures in Computer Science*, 24(4), 2014.

2 Clovis Eberhart and Tom Hirschowitz. Presheaves for Processes and Unfoldings. `http://www.lama.univ-savoie.fr/~eberhart/ProcUnfold.pdf`. Online draft.

3 Tom Hirschowitz. Full abstraction for fair testing in CCS (expanded version). *Logical Methods in Computer Science*, 10(4), 2014.