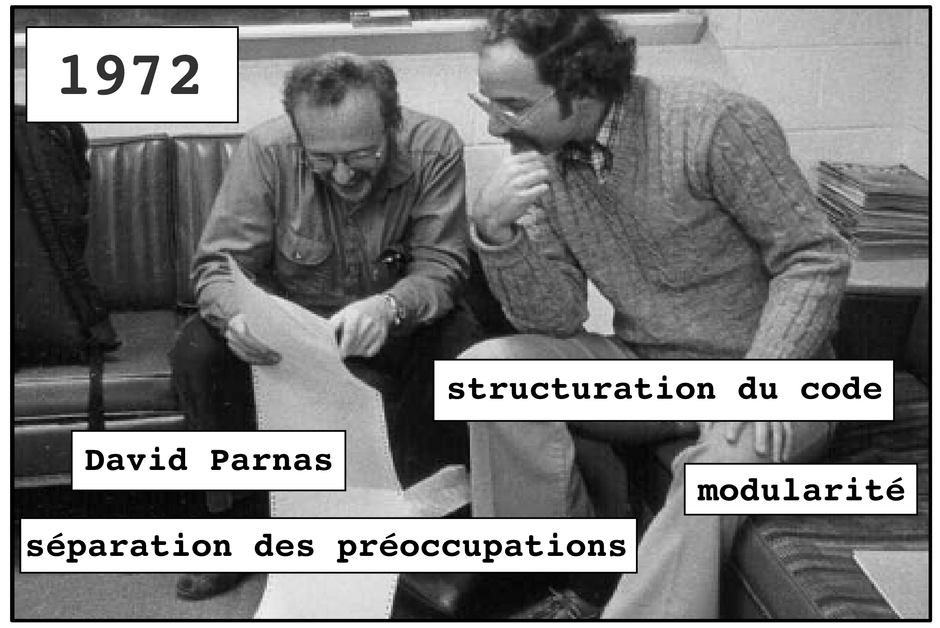
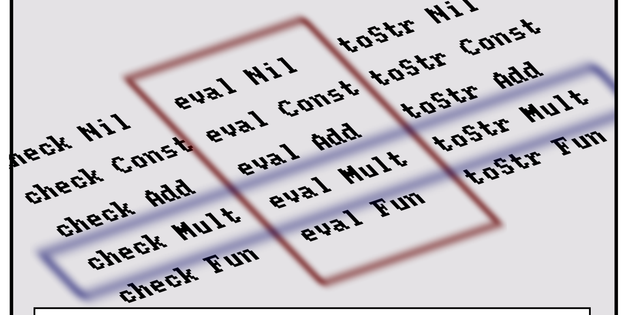
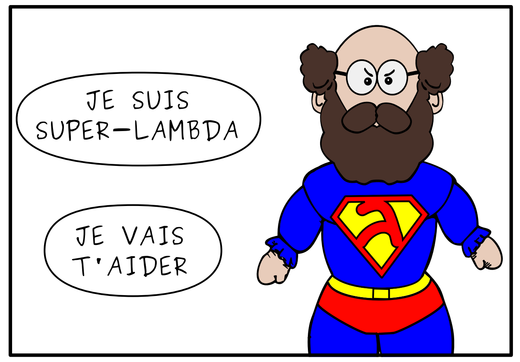
Tyrannie

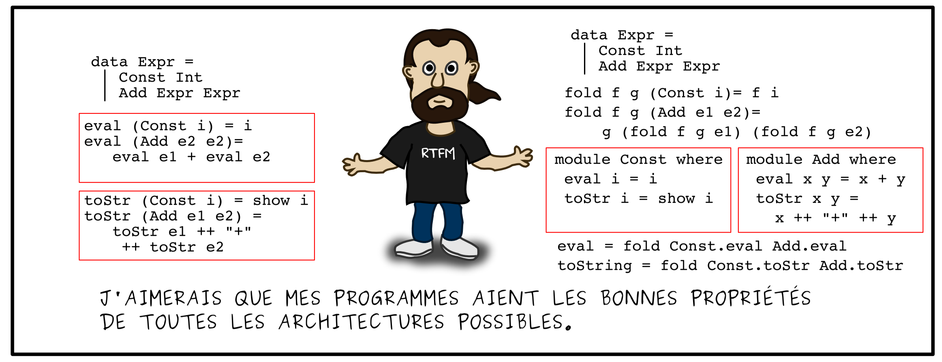


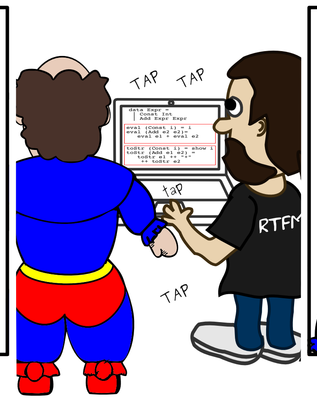
Les choix de conception tendent à favoriser une décomposition "dominante"

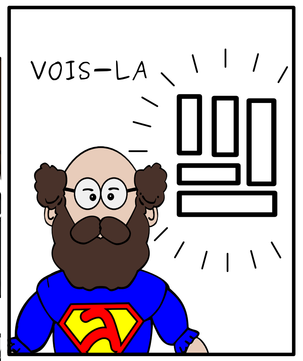


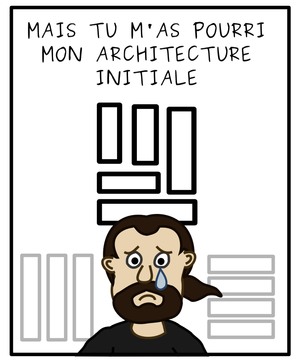
...ce qui affaiblit les décompositions secondaires

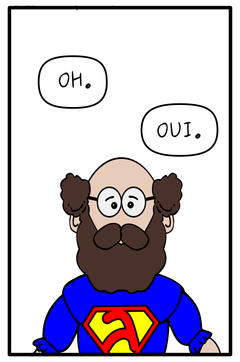






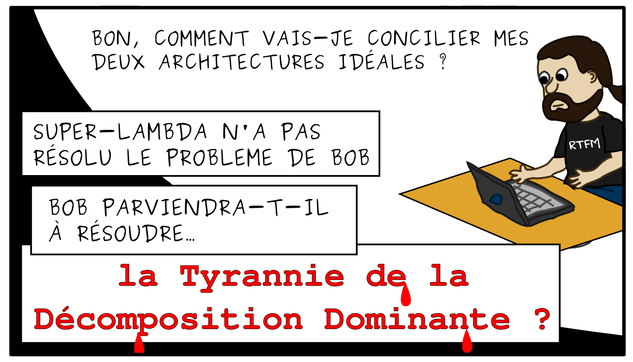




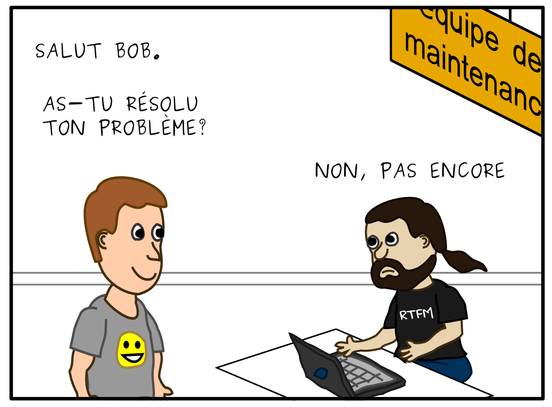


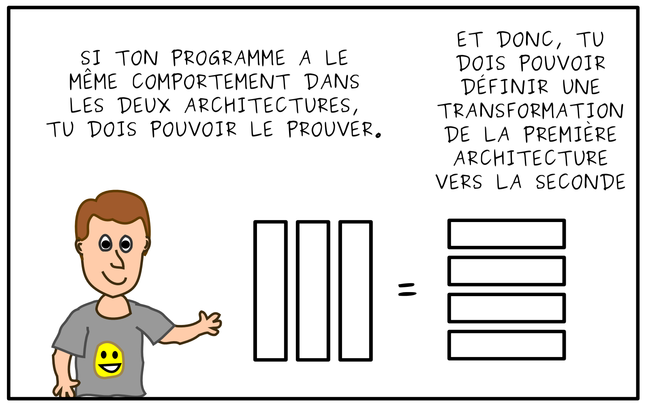
NON.

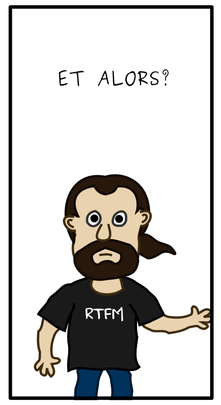
PAS SUR MOI

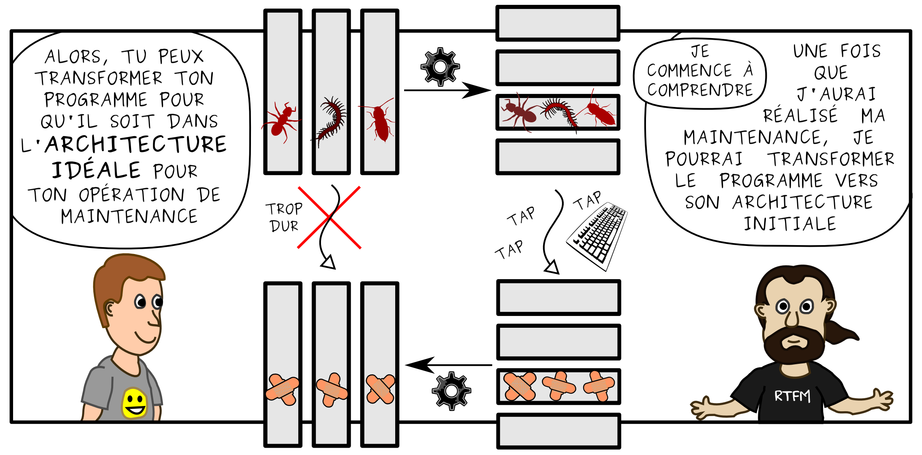


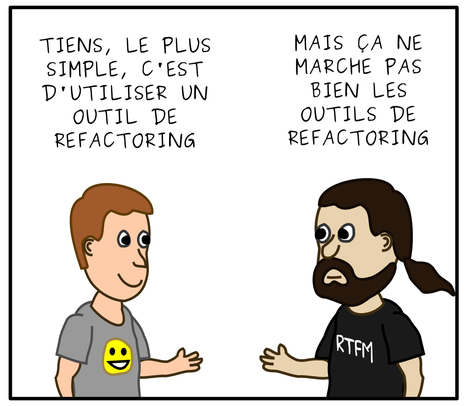
Transformations

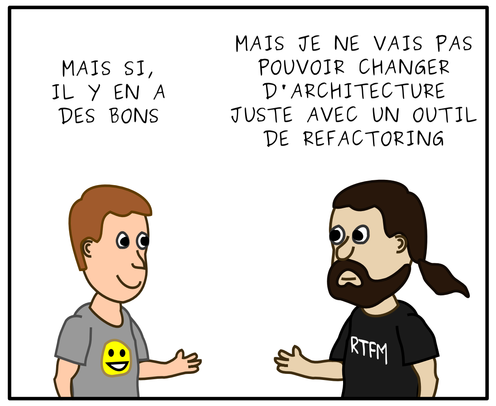


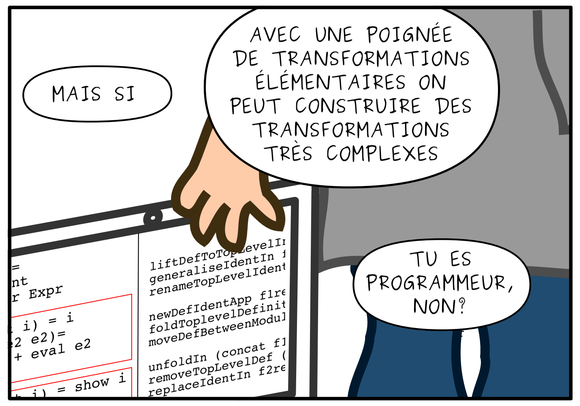


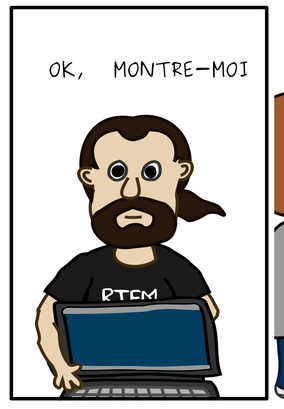


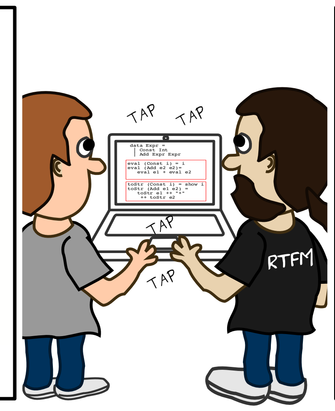


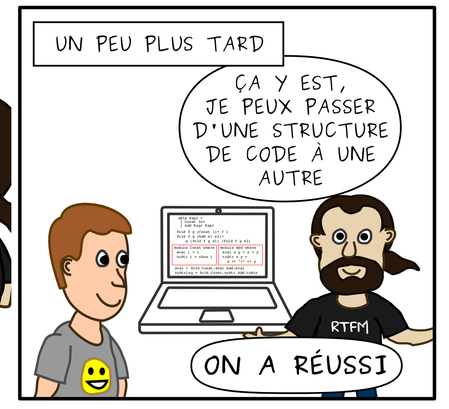


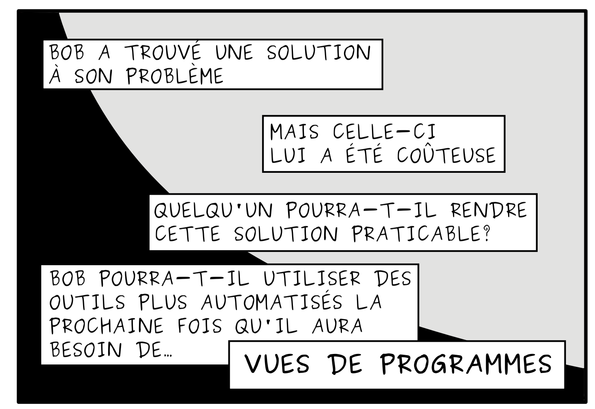


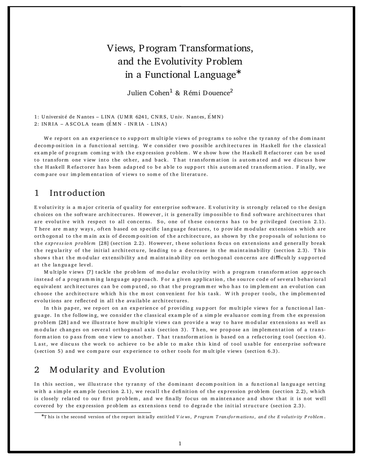












Views, Program Transformations, and the Evolutivity Problem in a Functional Language*

Julien Cohen[1] & Rémi Douence[2]

1- Université de Nantes – LINA (UMR 6241, CNRS, Univ. Nantes, EMN)
2- INRIA – ASCOLA team (EMN - INRIA - LINA)

We report on an experience to support multiple views of programs to solve the tyranny of the dominant decomposition in a functional language. We study the definition of the expression problem (section 2.1), which is closely related to our first problem, and we finally focus on maintenance and show that it is not well covered by the expression problem as extensions tend to degrade the initial structure (section 2.2).

1 Introduction

2 Modularity and Evolution

Références

[1] Julien Cohen and Rémi Douence. Views, Program Transformations, and the Evolutivity Problem in a Functional Language. Research Report hal-00481441, version 2, January 2011.

[2] Huiqing Li and Simon Thompson. Tool Support for Refactoring Functional Programs. In Partial Evaluation and Program Manipulation, Jan. 2008.

[3] David L. Parnas. On the criteria to be used in decomposing systems into modules. Commun. ACM, 15:1053–1058, 1972.

[4] Peri Tarr, Harold Ossher, William Harrison, and Stanley M. Jr. Sutton. N degrees of separation: multi-dimensional separation of concerns. In ICSE, 1999. ACM.

[5] Philip Wadler. The expression problem. Note to Java Genericity mailing list, Nov. 1998.

[6] Philip Wadler. Views: A way for pattern matching to cohabit with data abstraction. In POPL, 1987.

[7] Lehman M M, Laws of Program Evolution – Rules and Tools for Programming Management, Proc. Infotech State of the Art Conf., Why Software Projects Fail, 1978.

Julien Cohen (Univ. Nantes, Ascola)
Rémi Douence (INRIA, Ascola)
Akram Ajouli (EMNantes, Ascola)
Mai 2011