How can OCL be improved? *(at least a missing feature)*

- **Why do we need to *improve* OCL?**
  - To make it widely used by industry in practice

- **Why is OCL (almost) not used in the industry?**
  - Industry does not build (UML) models – *there is still a long way to go*
  - OCL is already too complex to be understood by industry people (and even by our students …) – *should we extend it further then?*

- **Some personal "concerns":**
  - OCL is a declarative language: is the iterate construct declarative?
  - *(OCL spec.)* "*everywhere in the UML specification where the term expression is used, an OCL expression can be used (e.g., invariants, pre and postconditions), but other placements are possible too*"
  - *When should I use tuple types ? Why are they useful for?*
    (which book should people read to learn OCL in practice?)

How to make OCL to be used by industry?

(in my opinion the best way to improve OCL)

- **We need tools to perform *things* automatically from OCL**

- **We should define “purpose-specific” subsets of OCL**
  - E.g. to define invariants, to define operation contracts, to define model transformations, etc.
  - Each of them defined in an incremental way, from the most common patterns to the most complex and particular issues.
  - We need books for people to make self-learning of OCL, for each specific subset.
  - Simple is better! (even if we miss some specific feature)