Context:
Legacy code of a dedicated tool handling domain specific data gathers valuable expertise. However, in many cases, this code must be rewritten in order to make it apply to semantically equivalent but incompatible data. This update can be complex and error-prone.

How to improve the reuse of legacy tools?

Approach: Automatic adaptation of models, instead of rewriting or adapting the tool itself
- Based on co-evolution operators (rename, remove, flatten, hide, etc.).
- Refactoring at metamodel-level.
- Migration round-trip at model-level.
  - Graph based model semantics.
  - Asymmetrical onward and reverse migrations.
  - Tool characterized by a dependency graph.

Round-trip and Example:

Application domain

Co-evolution

Legacy tool domain

Expected Tool Metamodel

Notation:

- instance #i
- scalar value
- reference
- attribute

Publications: