

References

1. Wimmer, M., Jouault, F., Cabot, J.: A catalogue of refactorings for model-to-model transformations. *Journal of Object Technology* 11(2), 2–1 (2012)

A Annexed

```
operation HTML!HTMLElement
removeStyle(pKey : Any) : Boolean{
  var found = false;
  for(style in self.styles){
    if(style.key == pKey){
      self.styles.remove(style);
      found = true;
    }
  }
  return found;
}
```

Listing 1. Restructuring: Imperative filtering

```
rule HTMLTable2NewsletterTable
transform
  htmlTable : HTML!Table
to
  table: Newsletter!Table{

    for(row in htmlTable.trs)
    {
      var newRow = new Newsletter!Row;
      ...
    }
  }
```

Listing 2. OCL Optimization: Imperative element creation in loops

Table 1. Catalog of bad smells proposed in literature and detected by our approach

Acronym[1]	Name	Problem	Algorithm
DOE	Duplicated and complex expressions	Some OCL expressions are duplicated or too complex. The latter results in less readable code and the former in less maintainable code	All calls to operations inside an OCL are counted and if the result is higher than a threshold, it is shown as a complex OCL. In this case we set it to 5 based on our academic experience
TOC	Trivial operations called once	Operations that are called once all over the transformation (for example, to create an element), make the code less understandable	The entire transformation is verified checking how many times each operation is used, in the case that the operation is called just once and the operation has less than a defined threshold of lines (excluding comments and whitespaces, we defined the threshold in 3 due to our dataset review) then the bad smell is identified
REB	Rule body embedded into blocks	The body of the output pattern is embedded into if (and else) blocks to separate the logic being applied in different cases, which makes transformation hard to understand	Each rule is verified and if the first element contained in the output pattern is an if, then the bad smell count is increased
TMB	Target meta-model attribute set in multiple bindings	A target feature is being set by at least 2 bindings, this makes the transformation complex and harder to understand	Check the body of each rule, looking if each target binding is set more than once, if that case is found this bad smell type is increased by one
NIC	Nested if/else chains replaced with switch	Nested IF/ELSE chains are commonly used to convert values from the source model into values needed for the target model, e.g., convert Java to SQL data types. This is discouraged as the switch statement is less complex and more suitable for this kind of scenarios	The discoverer checks each statement and if it finds an else if this bad smell is increased by one
CSF	A chain of selecting/first in OCL using selectFirst	An OCL expression containing Select/First operation chains is difficult to read and unnecessarily expensive calculated	All OCL expressions are checked searching for the combination of the operations: select and first, if there is at least one, then this bad smell is increased by one

Table 2. Distribution of bad smells across datasets

Ocurrences	Github	Uniandes	York	Total
<i>0</i>	15.0	87.0	16.0	118.0
<i>1 to 5</i>	25.0	73.0	6.0	104.0
<i>6 to 10</i>	11.0	14.0	0.0	25.0
<i>More than 10</i>	21.0	17.0	1.0	39.0
<i>Rate by Transformation</i>	8.2	3.3	1.0	4.4

Table 3. Table of ETL transformation metrics

Acronym	Name	Description
MR	Matched Rules	Number of matched rules.
LR	Lazy Rules	Number of lazy rules.
CLR	Calls to lazy rule per rule	Average of invocations to lazy rule per rule
OWC	Operations with context	Number of operations with context, it means that the operation is specified to a particular type
ONC	Operations without context	Number of operations without context, it means that the operations can be used by any type
COR	Calls to operations per rule	Average of invocations to operations per rule
NIF	Number of ifs	Total number of the if statements found in the code
NI	Number of iterators	Total number of iterators found in the code
VPR	Variables per rule	Average of new variable being defined per rule
UWO	Unused Operations	Total number of operations defined, but not used in the transformation
UUP	Unused Parameters	Total number of parameters defined in operations, but not used in it