

Evaluation

In this document, we describe the experiments we have performed in order to evaluate the *functional suitability* of our proposal through the use of our reasoning-based prototype available at [SMEP]. Functional suitability is one of the characteristics included in the ISO/IEC 25010 - System and Software Quality Requirements and Evaluation (SQuaRE) standard [SQuaRE] which, at the same time, is further subdivided into three quality sub-characteristics: *functional appropriateness*, *functional correctness* and *functional completeness*.

Taking this into account, we have evaluated these characteristics by applying our prototype to four real KPI systems (denoted as CS1, CS2, CS3 and CS4, respectively¹). More specifically, we have evaluated: 1) *functional suitability*, in general, by means of CS1, 2) *functional appropriateness*, by means of CS2, 3) *functional correctness*, by means of CS3, and 4) *functional completeness*, for which, we have used CS4.

Considerations:

- For simplicity, in these experiments we have considered that the person in charge has decided to apply all traceability aspects (consistency conditions and propagation aspects are always applied in the prototype's transformations). Thus, the Formula transformation invocations include `true` values for parameters associated to decisions T1, T3 and T4. In the case of T2 (storage of the motivation), the motivation itself is passed as a string.
- We note that the KPIs included in the four case studies do not include target elements. For this reason, we have considered targets on our own in all KPIs.
- The experiments have been performed by using the *Formula* command line interface [Form].

¹ CS refers to Case Study.

Case study CS1

Reference: J. Bastos, R. Garcia, F. Freire, Indicators for waste prevention and management - measuring circularity, 2019. UrbanWINS project- Innovative Strategic Plans for Urban Waste Reduction and Management. Online at https://edisciplinas.usp.br/pluginfile.php/7005693/mod_resource/content/1/d2.3-Indicators-for-waste-prevention-and-management.pdf². Last visited on September, 2024.

Evaluated characteristic: *functional suitability*, which represents the capability of a product to provide functions that meet stated and implied needs of intended users [SQuaRE].

Experiments' goal. We have evaluated this property by applying the prototype to several indicators defined in the report of CSI and proving that the evolution information produced by our tool is detailed enough and useful to track the performed changes.

This case study refers to a report that presents a set of indicators on circular economy, waste prevention and management. The report has been developed within the UrbanWINS- Innovative Strategic Plans for Urban Waste Reduction and Management- project. More specifically, it presents 35 *waste indicators* (identified with numbers from 1 to 35) and 25 *circular economy indicators* (identified with numbers from 36 to 60). From those indicators, we have identified 14 atomic and 46 compound indicators.

As advanced previously, we have used this case study to assess the wealth and usefulness of the evolution information provided by our prototype. More specifically, we have based on several circular economy indicators from those defined in the report and performed several changes to check that the evolution information produced by the prototype (regarding tracing, consistency conditions and propagation aspects) is useful to track the performed changes.

Source KPIs (identified as stated in the report)

NO.	Granularity	Indicator	Calculation	Description
48	Atomic	Imports (Imp)	Mass amount of goods entering the geographic area	Goods entering the economy/territorial unit.
46	Atomic	Exports (Exp)	Mass amount of goods leaving the geographic area	Goods leaving an economy/territorial unit. Can be used to describe the support provided by an urban area to other areas; displays the role of the city in meeting the needs of other system.
41	Atomic	Domestic extraction (DE)	Mass amount of raw material extracted from the natural environment	Input from the natural environment to be used in the economy/territorial unit. DE is the annual amount of raw material (except for water and air) extracted from the natural environment.
40	Compound	Direct material input (DMI)	$DMI = DE + Imp$	Measures the direct input of materials for use into the economy. It can be used to describe the total material needs of an urban area.
42	Compound	Domestic material consumption (DMC)	$DMC = DMI - Exp$	Measures the total amount of material directly used in an economy (excluding indirect flows). It provides the effective quantities of goods consumed in a given area
38	Compound	Dependency on other systems (Dep)	$Dep = Imp/DMI$	This indicates the share of DMI that is from imports. It provides insight on the vulnerability of an urban area the extent to which it is dependent on other outer areas.

² Previously available at <https://www.urbanwins.eu/wp-content/uploads/2019/05/d2.3-Indicators-for-waste-prevention-and-management.pdf>

We note that the source KPI system consist of three atomic indicators and three compounds.

KPI System in Formula

```
// KPIImp:
KPIImp is KPI(1,"KPIImp",Active,0)
KPIFormula(1,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(1,"", "",Atomic),KPIImp)
basicValue(KPIImp,2)
// KPIExp:
KPIExp is KPI(2,"KPIExp",Active,0)
KPIFormula(2,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(2,"", "",Atomic),KPIExp)
basicValue(KPIExp,3)
// KPIDE:
KPIDE is KPI(3,"KPIDE",Active,0)
KPIFormula(3,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(3,"", "",Atomic),KPIDE)
basicValue(KPIDE,4)
//KPIDMI=KPIDE+KPIImp
KPIDMI is KPI(4,"KPIDMI",Active,0)
KPIFormula(4,"KPIDE+KPIImp", "+",Compound)
KPIElement_belongs_to_Version(KPIFormula(4,"KPIDE+KPIImp", "+",Compound), KPIDMI)
is_based_on(KPIDMI,KPIDE)
is_based_on(KPIDMI,KPIImp)
// KPIDMC = KPIDMI - KPIExp
KPIDMC is KPI(5,"KPIDMC",Active,0)
KPIFormula(5,"KPIDMI-KPIExp", "-",Compound)
KPIElement_belongs_to_Version(KPIFormula(5,"KPIDMI-KPIExp", "-",Compound), KPIDMC)
is_based_on(KPIDMC,KPIDMI)
is_based_on(KPIDMC,KPIExp)
// KPIDep = KPIImp/KPIDMI
KPIDep is KPI(6,"KPIDep",Active,0)
KPIFormula(6,"KPIImp/KPIDMI", "/",Compound)
KPIElement_belongs_to_Version(KPIFormula(6,"KPIImp/KPIDMI", "/",Compound), KPIDep)
is_based_on(KPIDep,KPIImp)
is_based_on(KPIDep,KPIDMI)
```

Starting from the previous indicators, we have considered the following two sequential evolution changes.

Experiment 1. Creation of the indicator PTB:

NO.	Granularity	Indicator	Calculation	Description
55	Compound	Physical trade balance (PTB)	PTB= Imp -Exp	Defined as the difference between physical imports and physical exports. A physical trade surplus indicates a net import of materials, whereas a physical trade deficit indicates a net export.

Starting from the previous KPIs, we have considered a new indicator, PTB, defined as $\text{Imp} - \text{Exp}$. Thus, we have applied the `CreateKPI` transformation, using suitable arguments.

Formula instructions³:

```
//load the specification
load path\AFFECTK\FormulaPrototype\FormulaPrototype.4ml
//Invoke the transformation
transform CreateKPI <"KPIPTB", 7, Compound, "KPIImp-KPIExp", lessThan, 78, -1, "-",
IN.KPI(1,"KPIImp",Active,0), IN.KPI(2,"KPIExp",Active,0),true, "new needs" > KPISystem
//Show the resulted KPI model
knows CreateKPI#1
```

Formula results:

As a result, Formula has shown that the indicator does not contradict any consistency condition from those implemented by the `CreateKPI` transformation (*nonRepeatedName*, *uniqueness*, and *coherence*). More specifically, the `CreateKPI.conformsTransform` query is evaluated as `true` since queries `CreateKPI.repeatedName`, `CreateKPI.nonUniqueness` and `CreateKPI.incoherence` are evaluated as `false`:

```
Query evaluation results:
CreateKPI.@user_conforms => True
CreateKPI.conformsTransform => True
CreateKPI.incoherence => False
CreateKPI.nonUniqueness => False
CreateKPI.repeatedName => False
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.nonUniqueness => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.nonUniqueness => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
```

³ Since target values are not specified we consider lessThan 5 for simplicity and KPITarget elements are not included in the results.

Additionally, it can be shown that neither the source KPI model (see queries preceded by IN) nor the target KPI model (see queries preceded by out) violate the constraints defined by the Formula domains.

The Formula report also shows the resulting KPI system (including the new KPI in Active state and with 0 as version), together with the corresponding instances related to evolution aspects.

```

1- PACreateKPI1 is PatternApplication("date","User","no comments",CreateKPI,"new needs",Ejecuted)
2- AppliedTraceOperation("",KPI(7,"KPIPTB",Active,0), PACreateKPI1)
3- AppliedPropagationOperation(
    PACreateKPI1,
    PatternApplication("date","User","no comments",UpdateCalculationRule,"InfluencedBy",
        InStudyOptional),
    "Check whether the application of the 'influences' create pattern to the indicated KPI may imply the
        need for applying the 'influenced by' update calculation rule pattern",
    KPI(7,"KPIPTB ",Active,0))
4- AppliedPropagationOperation(
    PACreateKPI1,
    PatternApplication("date","User","no comments", DeleteKPI, "InfluencedBy", InStudyOptional),
    "Check whether the application of the 'influences' create pattern to the indicated KPI may imply the
        need for applying the 'influenced by' delete pattern",
    KPI(7,"KPIPTB ",Active,0))
5- AppliedPropagationOperation(
    PACreateKPI1,
    PatternApplication("date","User","no comments", CreateKPI, "InfluencedBy", InStudyOptional),
    "Check whether the application of the 'influences' create pattern to the indicated KPI may imply the
        need for applying the 'influenced by' creation pattern",
    KPI(7,"KPIPTB ",Active,0))
6- //KPIs definitions, included the new one
    KPIPTB is KPI(7,"KPIPTB",Active,0)
    KPIFormula(7,"KPIImp-KPIImp","-",Compound)
    KPIElement_belongs_to_Version( KPIFormula(7,"KPIImp-KPIExp","-",Compound), KPIPTB)
    is_based_on(KPIPTB,KPIImp)
    is_based_on(KPIPTB,KPIImp)
    ....
    //Derived calculatedValues, relatedKPIs, KPIValue instances, etc.

```

In this report, we can see that all constraints are satisfied (no `AppliedConsistencyCondition` instances are created with the violated constraints).

Additionally, and according to the CreateKPI pattern, the transformation execution informs the user (by means of suitable instances of `AppliedPropagationOperation`) about the propagation actions she/he could take, choosing between: i) modifying the calculation rules of other KPIs as appropriate (line 3), making use of the new KPI, and/or ii) deleting or creating other KPIs, considering the creation of the new KPI (lines 4 and 5, respectively). In this case, since such propagation actions are not needed to maintain the consistency of the KPI system, they are identified as `InStudyOptional`.

Later, based on such suggested propagations and the considered KPI system, we have decided to redefine the indicator DMC so that it includes PTB ($DMC = DE + PTB$), by applying the `UpdateCalculationRule` transformation.

Formula instructions:

```
load path\AFFECTK\FormulaPrototype\FormulaPrototype.4ml

transform UpdateCalculationRule < IN.KPI(5,"KPIDMC",Active,0), -1, "KPIDE+KPIPTB",
"+",IN.KPI(3,"KPIDE",Active,0), IN.KPI(7,"KPIPTB",Active,0), true,"Change DMC", true, true> KPISystem
```

As a result, Formula shows that the consistency conditions are satisfied (we note, for example, query `UpdateCalculationRule.conformsTransform => true`, showing that the queries stated by the transformation are evaluated as `false`, and that the `conforms` queries related to both source and target KPI models are also evaluated to `true`:

Query evaluation results:

```
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
IN.KPIMetamodel.nonUniqueness => False
UpdateCalculationRule.@user_conforms => True
UpdateCalculationRule.incoherence => False
UpdateCalculationRule.conformsTransform => True
UpdateCalculationRule.nonActiveState => False
UpdateCalculationRule.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
out.KPIMetamodel.nonUniqueness => False
```

In this case, in addition to the reporting information regarding traceability, consistency conditions and propagation aspects, we can also see the evolution of indicator DMC as follows:

- 1- **PAUpdateCR1** is **PatternApplication**("date","User","no comments",UpdateCalculationRule,"change DMC",Ejecuted)
- 2- **AppliedTraceOperation**(
 KPIFormula(5,"KPIDMI-KPIExp","-",Compound), KPIFormula(6,"KPIDE+KPIPTB","+",Compound),
 PatternApplication("date","User","no comments",
 PAUpdateCR1)
- 3- **AppliedPropagationOperation**(
 PAUpdateCR1,
 PatternApplication("date","User","no comments",UpdateTarget,"InfluencedBy",**InStudyCompulsory**),
 "Check whether the application of the 'influences' update calculation rule pattern to the indicated KPI
 may imply the need for applying the 'influenced by' update target pattern",
 KPI(5,"KPIDMC",Active,1))
- 4- **//KPIs definitions, included the previous and the new version of the indicator DMC**
 KPI(5,"KPIDMC",Deprecated,0)

```

KPIFormula(5,"KPIDMI-KPIExp","-",Compound)
KPIElement_belongs_to_Version(KPIFormula(5,"KPIDMI-KPIExp","-",Compound),KPI(5,"KPIDMC",Deprecated,0))
is_based_on(KPI(5,"KPIDMC",Deprecated,0),KPI(2,"KPIExp",Active,0))
is_based_on(KPI(5,"KPIDMC",Deprecated,0),KPI(4,"KPIDMI",Active,0))
KPI(5,"KPIDMC",Active,1)
KPIFormula(6,"KPIDE+KPIPTB","+ ",Compound)
KPIElement_belongs_to_Version(KPIFormula(6,"KPIDE+KPIPTB","+ ",Compound),KPI(5,"KPIDMC",Active,1))
is_based_on(KPI(5,"KPIDMC",Active,1),KPI(3,"KPIDE",Active,0))
is_based_on(KPI(5,"KPIDMC",Active,1),KPI(7,"KPIPTB",Active,0))

.....
//Derived calculatedValues, relatedKPIs relationships and KPIValue instances

```

As it can be seen in line 3, the application of the `UpdateTarget` transformation is indicated as propagated compulsory action in order to check whether the new calculation rule is coherent with the target or status options of the KPIDMC. As advanced previously, since target aspects are not considered in the definition of the UrbanWINS project indicators, we have not considered this element of KPIs in our experiments.

We also note that, since there is no other KPIs that include indicator DMC in their formulas, the report does not include a `AppliedPropagationOperation` instance to compulsorily propagate a modification of the calculation rule of other indicators.

We also note how both versions of KPI DMC are included: the version previous to the evolution change with “Deprecated” state and version identifier 0, and the current one with “Active” state and version identifier 1.

Experiment 2. Deletion of the indicator DMI:

We have supposed that the indicator DMI is not longer needed in the system and we have applied the `DeleteKPI` transformation.

Formula instructions:

```
load path to FormulaPrototype.4ml
transform DeleteKPI<IN.KPI(4,"KPIDMI",Active,0),true, "no longer collected", true, true> KPISystem
```

Formula results:

As a result, the Formula console has informed us that the *coherence* condition is not satisfied (`incoherence` Formula constraint is true) since there is an indicator that depends on indicator DMI (more specifically, indicator Dep).

```
Query evaluation results:
DeleteKPI.@user_conforms => False
DeleteKPI.incoherence => True
DeleteKPI.conformsTransform => False
DeleteKPI.nonActiveState => False
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
IN.KPIMetamodel.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
out.KPIMetamodel.nonUniqueness => False
```

The transformation report also shows (by means of the corresponding `AppliedConsistencyCondition` instance) that the query is not satisfied.

Consequently, the transformation report also indicates (by means of the generated `AppliedPropagationOperation` instances) to choose among different compulsory options to maintain the system consistency: i) modifying the calculation rules of the KPI(s) that include the KPI to be deleted (indicator Dep), so that they do not contain the KPI (line 4), or ii) deleting the KPIs that include the KPI to be deleted (line 5). Both actions are identified as `InStudyCompulsory`. Additionally, it suggests the person in charge to assess if she/he wants to delete or create other KPIs, considering the deletion of the KPI (lines 6 and 7). These actions are identified as `InStudyOptional`.


```

1- PADeleteKPI1 is PatternApplication("date","User","no comments", DeleteKPI,"no longer
collected",Ejecuted)
2- AppliedTraceOperation(
    KPI(4,"KPIDMI",Deprecated,0),
    KPI(4,"KPIDMI",Deleted,1),
    PADeleteKPI1)
3- AppliedConsistencyCondition(PADeleteKPI1, KPI(4,"KPIDMI",Active,0),"incoherence")
4- AppliedPropagationOperation(
    PADeleteKPI1,
    PatternApplication("date","User","no comments", UpdateCalculationRule, "InfluencedBy",
InStudyCompulsory),
    "Calculation rule of the indicated KPI would need to be checked",
    KPI(6,"KPIDep",Active,0))
5- AppliedPropagationOperation(
    PADeleteKPI1,
    PatternApplication("date","User","no comments",DeleteKPI,"InfluencedBy",InStudyCompulsory),
    "The indicated KPI could need to be deleted",
    KPI(6,"KPIDep",Active,0))
6- AppliedPropagationOperation(
    PADeleteKPI1,
    PatternApplication("date","User","no comments",CreateKPI,"InfluencedBy",InStudyOptional),
    "Check whether the application of the 'influences' delete pattern to the indicated KPI (in the
Deleted state) may imply the need for applying the 'influenced by' create pattern",
    KPI(4,"KPIDMI",Deleted,1))
7- AppliedPropagationOperation(
    PADeleteKPI1,
    PatternApplication("date","User","no comments",DeleteKPI,"InfluencedBy",InStudyOptional),
    "Check whether the application of the 'influences' delete pattern to the indicated KPI (in the
Deleted state) may imply the need for applying the 'influenced by' delete pattern",
    KPI(4,"KPIDMI",Deleted,1))
8- //KPIs definitions, included the previous and the new version of the indicator DMI
KPI(4,"KPIDMI",Deleted,1)
KPIFormula(4,"KPIDE+KPIImp","+",Compound)
KPIElement_belongs_to_Version(KPIFormula(4,"KPIDE+KPIImp","+",Compound),KPI(4,"KPIDMI",Deleted,1))
is_based_on(KPI(4,"KPIDMI",Deleted,1),KPI(1,"KPIImp",Active,0))
is_based_on(KPI(4,"KPIDMI",Deleted,1),KPI(3,"KPIDE",Active,0))
KPI(4,"KPIDMI",Deprecated,0)
KPIElement_belongs_to_Version(KPIFormula(4,"KPIDE+KPIImp","+",Compound),KPI(4,"KPIDMI",Deprecated,0))
is_based_on(KPI(4,"KPIDMI",Deprecated,0),KPI(1,"KPIImp",Active,0))
is_based_on(KPI(4,"KPIDMI",Deprecated,0),KPI(3,"KPIDE",Active,0))....
//Derived calculatedValues, relatedKPIs, KPIValue instances, etc.

```

In particular, we have decided to update indicator Dep as propagated change.

```

load path to FormulaPrototype.4ml
transform DeleteKPI<IN.KPI(6,"KPIDep",Active,0),true,"propagated deletion", true, true> KPISystem

```

Formula results:

As a result, the Formula console has informed us that all condition are satisfied.

Query evaluation results:

```
DeleteKPI.@user_conforms => True
DeleteKPI.incoherence => False
DeleteKPI.conformsTransform => True
DeleteKPI.nonActiveState => True
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
IN.KPIMetamodel.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
out.KPIMetamodel.nonUniqueness => False
```

The resulted report is:

- 1- **PADeleteKPI2** is **PatternApplication**("date","User","no comments",DeleteKPI,"propagated deletion",Ejecuted)
- 2- **AppliedTraceOperation**(
KPI(6,"KPIDep",Deprecated,0),
KPI(6,"KPIDep",Deleted,1),
PADeleteKPI2)
- 3- **AppliedPropagationOperation**(
PADeleteKPI2,
PatternApplication("date","User","no comments",CreateKPI,"InfluencedBy",**InStudyOptional**),
"Check whether the application of the 'influences' delete pattern to the indicated KPI (in the Deleted state) may imply the need for applying the 'influenced by' create pattern",
KPI(6,"KPIDep",Deleted,1))
- 4- **AppliedPropagationOperation**(
PADeleteKPI2,
PatternApplication("date","User","no comments",DeleteKPI,"InfluencedBy",**InStudyOptional**),
"Check whether the application of the 'influences' delete pattern to the indicated KPI (in the Deleted state) may imply the need for applying the 'influenced by' delete pattern",
KPI(6,"KPIDep",Deleted,1))

```

5- //KPIs definitions, included the previous and the new version of the KPIDMI
KPI(4,"KPIDMI",Deleted,1)
KPIFormula(4,"KPIDE+KPIImp","+",Compound)
KPIElement_belongs_to_Version(KPIFormula(4,"KPIDE+KPIImp","+",Compound),KPI(4,"KPIDMI",Deleted,1))
is_based_on(KPI(4,"KPIDMI",Deleted,1),KPI(1,"KPIImp",Active,0))
is_based_on(KPI(4,"KPIDMI",Deleted,1),KPI(3,"KPIDE",Active,0))

KPI(4,"KPIDMI",Deprecated,0)
KPIElement_belongs_to_Version(KPIFormula(4,"KPIDE+KPIImp","+",Compound),KPI(4,"KPIDMI",Deprecated,0))
is_based_on(KPI(4,"KPIDMI",Deprecated,0),KPI(1,"KPIImp",Active,0))
is_based_on(KPI(4,"KPIDMI",Deprecated,0),KPI(3,"KPIDE",Active,0))
.....

KPI(6,"KPIDep",Deleted,1)
KPIFormula(6,"KPIImp/KPIDMI","/",Compound)
is_based_on(KPI(6,"KPIDep",Deleted,1),KPI(1,"KPIImp",Active,0))
is_based_on(KPI(6,"KPIDep",Deleted,1),KPI(4,"KPIDMI",Deleted,1))
KPIElement_belongs_to_Version(KPIFormula(6,"KPIImp/KPIDMI","/",Compound),KPI(6,"KPIDep",Deleted,1))

KPI(6,"KPIDep",Deprecated,0)
is_based_on(KPI(6,"KPIDep",Deprecated,0),KPI(1,"KPIImp",Active,0))
is_based_on(KPI(6,"KPIDep",Deprecated,0),KPI(4,"KPIDMI",Deleted,1))
KPIElement_belongs_to_Version(KPIFormula(6,"KPIImp/KPIDMI","/",Compound),KPI(6,"KPIDep",Deprecated,0))

//Derived calculatedValues, relatedKPIs, KPIValue instances, etc.

```

Since there is no indicator that depends on Dep, no compulsory propagation actions are indicated. The new version of the indicator `KPIDep` is also shown in the report.

Case study CS2

Reference: IAPT, Improving Access to Psychological Therapies, Key Performance Indicators (IAPT KPIs), 2014. Online at <https://digital.nhs.uk/data-and-information/publications/statistical/improving-access-to-psychological-therapies-key-performance-indicators-iapt-kpis>. Last visited on September, 2024.

Evaluated characteristic: *functional appropriateness*, which represents the capability of a product to provide functions that facilitate the accomplishment of specified tasks and objectives [SQuaRE].

Experiments' goal. We have evaluated the feasibility of the proposal for managing evolution by performing the real evolution changes that the IAPT's indicators.

In this case study we use the KPIs officially used in the IAPT (Improving Access to Psychological Therapies, 2014) programme. This program was established in 2008 in order to improve access to Psychological Therapies for people with Depression and Anxiety Disorders. Since then, the program has defined a family of several indicators which have evolved along time. These indicators are composed by KPIs and Headline indicators as an agreed mechanism for measuring patients' progress: while KPIs are at the tactical level of management and control performance, Headline indicators (we refer to them as HIs) are strategic indicators used to support change decisions (they can also be considered as atomic and compound, respectively).

Here, we present the indicators included in the different nine versions published at⁴, as well as the evolution changes they have undergone.

Atomic indicators (KPIs)

1. **KPI1:** Level of Need. It presents the number of people who have depression and/or anxiety disorders in the general adult population. The number presenting population is produced as a result of the Psychiatric Morbidity Survey.

Evolution aspects: This indicator is collected from the first version until the last one.

2. **KPI3a:** The number of people who have been referred for psychological therapies during the reporting quarter.

Evolution aspects: This indicator is collected from the first version until the last one.

3. **KPI3b:** The number of active referrals who have waited more than 28 days from referral to first treatment/first therapeutic session (at the end of the reporting quarter).

Evolution aspects: This indicator is collected from the first version until the last one.

4. **KPI4:** The number of people who have entered (i.e. received) psychological treatment, (i.e. had their first therapeutic session) during the reported quarter is related to the concept person.

Evolution aspects: This indicator is collected from the first version until the last one.

⁴ <https://digital.nhs.uk/data-and-information/publications/statistical/improving-access-to-psychological-therapies-key-performance-indicators-iapt-kpis>

5. **KPI5:** The number of people who have completed treatment (minimum 2 treatment contacts) during the reporting quarter.

Evolution aspects: This indicator is collected from the first version until the last one.

6. **KPI6a:** The number of people who are “moving to recovery” (of those who have completed treatment, those who at initial assessment achieved "caseness" and at final session did not) during the reporting period.

Evolution aspects: This indicator is collected from the first version until the last one.

7. **KPI6b:** The number of people who have completed treatment not at clinical caseness at initial assessment.

Evolution aspects: This indicator is collected from the first version until the last one.

8. **KPI7:** The number of people moving off sick pay or benefits during the reporting period.

Evolution aspects: This indicator is collected from the first version until the last one.

Compound indicators (Headline indicators)

9. **HI1(SQU16_04- PHQ13_05): Access Rate.** It indicates the people who have entered (i.e. received) treatment as a proportion of people with anxiety or depression. That is, the number of people entering treatment (KPI 4) over the level of need, i.e. the number of people with depression and anxiety disorders in the population (KPI1). It was expressed as $(KPI4/KPI1)*100$.

Evolution aspects:

- 1) From the first version until the fourth one (included) this measure was collected by means of the indicator with name **SQU16_04**. The indicator **SQU16_04** was no longer collected since the fifth version.
- 2) In the fifth version a new indicator with name **PHQ13_05** was created with the same formula as **SQU16_04**. This indicator remains from then until the last version.

10. **HI2 (Recovery rate- PHQ13_06): Recovery Rate.** The number of people not at caseness at their last session, as a proportion of people who were at caseness at their first session. That is, the number of people who are moving to recovery (KPI6), divided by the number of people who have completed treatment (KPI5) minus the number of people who have completed treatment that were not at caseness at initial assessment (KPI6b). It is expressed as $(KPI6a/(KPI5-KPI6b))*100$.

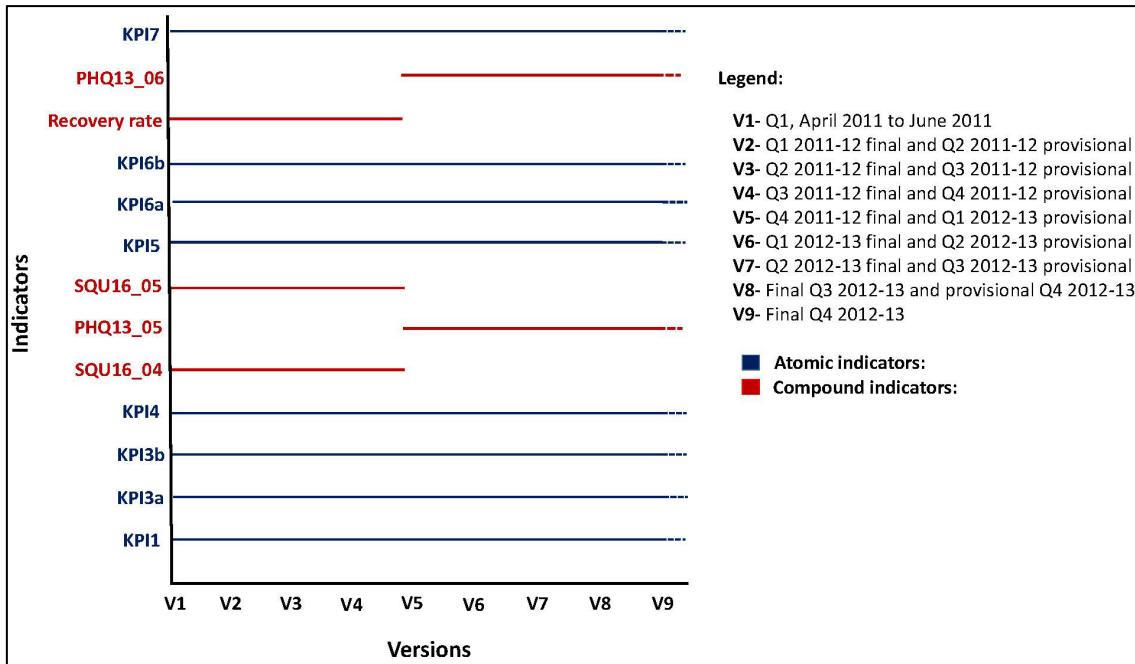
Evolution aspects:

- 1) From the first version until the fourth one (included) this measure was collected by means of the indicator with name **Recovery rate**. The indicator **Recovery rate** was no longer collected since the fifth version.
- 2) In the fifth version a new indicator with name **PHQ13_06** was created with the same formula as **Recovery rate**. This indicator remains from then until the last version.

11. **HI3 (SQU16_05):** This indicator is referred to as **SQU16_05**. The proportion of referrals that are entering treatment. It is expressed as $KPI4/KPI3a$.

Evolution aspects: This indicator is collected from the first version until the fourth one (included), from which it is no longer collected.

In order to illustrate the evolution changes along time, we next include a table with this information:



As advanced previously, we have used this case study to evaluate the capacity of our proposal to manage evolution. More specifically, we have performed the evolution of the indicators defined by the IAPT programme along the different nine published versions.

Evolution management transformations⁵

- 1) **First version- Creation transformation instructions:** We create from scratch the indicators included in the first version (atomic and compound indicators).

Atomic indicators

```
load path\AFFECTK\FormulaPrototype\FormulaPrototype.4ml
transform CreateKPI <"KPI1", 1, Atomic, "", lessThan, 5, 2, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI3a", 2, Atomic, "", lessThan, 5, 4, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI3b", 3, Atomic, "", lessThan, 5, 2, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI4", 4, Atomic, "", lessThan, 5, 4, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI5", 5, Atomic, "", lessThan, 5, 2, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI6a", 61, Atomic, "", lessThan, 5, 4, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI6b", 62, Atomic, "", lessThan, 5, 2, "", "", "", true, "initial version" > KPISystem
transform CreateKPI <"KPI7", 7, Atomic, "", lessThan, 5, 4, "", "", "", true, "initial version" > KPISystem
```

Result (formatted, without including target, tracing/propagation/consistency aspects)

```
// KPI1:
KPI1 is KPI(1, "KPI1", Active, 0)
KPIFormula(1, "", "", Atomic)
KPIElement_belongs_to_Version(KPIFormula(1, "", "", Atomic), KPI1)
basicValue(KPI1, 2)
```

⁵ As in the previous case study, since target values are not specified we consider lessThan 5 for simplicity and KPITarget elements are not included in the results.

```

// KPI3a:
KPI3a is KPI(2,"KPI3a",Active,0)
KPIFormula(2,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(2,"", "",Atomic),KPI3a)
basicValue(KPI3a,4)
// KPI3b:
KPI3b is KPI(3,"KPI3b",Active,0)
KPIFormula(3,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(3,"", "",Atomic),KPI3b)
basicValue(KPI3b,2)
// KPI4:
KPI4 is KPI(4,"KPI4",Active,0)
KPIFormula(4,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(4,"", "",Atomic),KPI4)
basicValue(KPI4,4)
// KPI5:
KPI5 is KPI(5,"KPI5",Active,0)
KPIFormula(5,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(5,"", "",Atomic),KPI5)
basicValue(KPI5,2)
// KPI6a:
KPI6a is KPI(61,"KPI6a",Active,0)
KPIFormula(61,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(61,"", "",Atomic),KPI6a)
basicValue(KPI6a,4)
// KPI6b:
KPI6b is KPI(62,"KPI6b",Active,0)
KPIFormula(62,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(62,"", "",Atomic),KPI6b)
basicValue(KPI6b,2)
// KPI7:
KPI7 is KPI(7,"KPI7",Active,0)
KPIFormula(7,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(7,"", "",Atomic),KPI7)
basicValue(KPI7,4)

```

Compound indicators

```

transform CreateKPI <"SQU16_04", 164, Compound, "KPI4/KPI1", lessThan, 5, -1,
"/",IN.KPI(4,"KPI4",Active,0),IN.KPI(1,"KPI1",Active,0),true, "initial version" > KPISystem
transform CreateKPI <"RecoveryRateA", 81, Compound, "KPI5-KPI6b", lessThan, 5, -1, "-
",IN.KPI(5,"KPI5",Active,0),IN.KPI(62,"KPI6b",Active,0),true, "initial version" > KPISystem
transform CreateKPI <"RecoveryRate", 8, Compound, "KPI6a/RecoveryRateA", lessThan, 5, -1,
"/",IN.KPI(4,"KPI6a",Active,0),IN.KPI(81,"RecoveryRateA",Active,0),true, "initial version" > KPISystem
transform CreateKPI <"SQU16_05", 165, Compound, "KPI4/KPI3a", lessThan, 78, -1,
"/",IN.KPI(4,"KPI4",Active,0),IN.KPI(2,"KPI3a",Active,0),true, "initial version" > KPISystem

```

Result (formatted, without including target, tracing/propagation/consistency aspects)

```

// SQU16_04: KPI4/KPI1.
SQU16_04 is KPI(164,"SQU16_04",Active,0)
KPIFormula(164,"KPI4/KPI1", "/",Compound)
KPIElement_belongs_to_Version(KPIFormula(164,"KPI4/KPI1", "/",Compound), SQU16_04)
is_based_on(SQU16_04,KPI4)
is_based_on(SQU16_04,KPI1)
// RecoveryRate: KPI6/(KPI5-KPI6b).
// RecoveryRatea=KPI5-KPI6b
RecoveryRatea is KPI(81,"RecoveryRatea",Active,0)
KPIFormula(81,"KPI5-KPI6b", "-",Compound)
KPIElement_belongs_to_Version(KPIFormula(81,"KPI5-KPI6b", "-",Compound), RecoveryRatea)
is_based_on(RecoveryRatea,KPI5)
is_based_on(RecoveryRatea,KPI6b)
//RecoveryRate = KPI6/ RecoveryRatea
RecoveryRate is KPI(8,"RecoveryRate",Active,0)
KPIFormula(8,"KPI6a/RecoveryRatea", "/",Compound)
KPIElement_belongs_to_Version(KPIFormula(8,"KPI6a/RecoveryRatea", "/",Compound),RecoveryRate)
is_based_on(RecoveryRate,KPI6a)
is_based_on(RecoveryRate,RecoveryRatea)
// SQU16_05: KPI4/KPI3a.
SQU16_05 is KPI(165,"SQU16_05",Active,0)
KPIFormula(165,"KPI4/KPI3a", "/",Compound)
KPIElement_belongs_to_Version(KPIFormula(165,"KPI4/KPI3a", "/",Compound), SQU16_05)
is_based_on(SQU16_05,KPI4)
is_based_on(SQU16_05,KPI3a)

```

2) Fifth version

Deletion transformation instruction to delete SQU16_04 and creation instructions to create PHQ13_05:

```
load path to FormulaPrototype.4ml

transform DeleteKPI<IN.KPI(164,"SQU16_04",Active,0), true, "no longer collected", true, true> KPISystem

transform CreateKPI <"PHQ13_05", 135, Compound, "KPI4/KPI1", lessThan, 5, -1,
"/",IN.KPI(4,"KPI4",Active,0),IN.KPI(1,"KPI1",Active,0),true, "version 5 need" > KPISystem
```

Result (formatted, without including target, tracing/propagation/consistency aspects)

```
// PHQ13_05: KPI4/KPI1.
PHQ13_05 is KPI(135,"PHQ13_05",Active,0)
KPIFormula(135,"KPI4/KPI1","/",Compound)
KPIElement_belongs_to_Version(KPIFormula(135,"KPI4/KPI1","/",Compound), PHQ13_05)
is_based_on(PHQ13_05,KPI4)
is_based_on(PHQ13_05,KPI1)
```

Deletion transformation instruction to delete SQU16_05:

```
load path to FormulaPrototype.4ml

transform DeleteKPI<IN.KPI(165,"SQU16_05",Active,0), true, "no longer collected", true, true> KPISystem
```

Deletion transformation instruction to delete RecoveryRate and creation instructions to create PHQ13_06:

```
load path to FormulaPrototype.4ml

transform DeleteKPI<IN.KPI(8,"RecoveryRate",Active,0), true, "no longer collected", true, true> KPISystem

transform CreateKPI <"PHQ13_06a", 1361, Compound, "KPI5-KPI6b", lessThan, 5, -1, "-
",IN.KPI(5,"KPI5",Active,0),IN.KPI(62,"KPI6b",Active,0),true, "Fifth version need" > KPISystem

transform CreateKPI <" PHQ13_06", 136, Compound, "KPI6a/PHQ13_06a", lessThan, 5, -1,
"/",IN.KPI(4,"KPI6a",Active,0),IN.KPI(81," PHQ13_06a",Active,0),true, "Fifth version need" > KPISystem
```

Result (formatted, without including target, tracing/propagation/consistency aspects)

```
// PHQ13_05: KPI6/(KPI5-KPI6b)
//PHQ13_05a=KPI5-KPI6b
PHQ13_05a is KPI(1351,"RecoveryRatea",Active,0)
KPIFormula(1351,"KPI5-KPI6b","-",Compound)
KPIElement_belongs_to_Version(KPIFormula(1351,"KPI5-KPI6b","-",Compound), PHQ13_05a)
is_based_on(PHQ13_05a,KPI5)
is_based_on(PHQ13_05a,KPI6b)
// PHQ13_05 = KPI6/ PHQ13_05a
PHQ13_05 is KPI(135,"RecoveryRate",Active,0)
KPIFormula(1358,"KPI6a/PHQ13_05a","/",Compound)
KPIElement_belongs_to_Version(KPIFormula(135,"KPI6a/PHQ13_05a","/",Compound),RecoveryRate)
is_based_on(RecoveryRate,KPI6a)
is_based_on(RecoveryRate,PHQ13_05a)
```

We obtain, as a result, the final version of the KPI system.

Case study CS3

Reference: Diamantini, Claudia and Potena, Domenico and Storti, Emanuele, SemPI: A Semantic Framework for the Collaborative Construction and Maintenance of a Shared Dictionary of Performance Indicators. Future Generation Computer Systems 54. DOI: 10.1016/j.future.2015.04.011.

Evaluated characteristic: *functional correctness*, which refers to the capability of a product to provide accurate results when used by intended users [SQuaRE].

Experiments' goal. Since, within our context, the main aspect related to correctness is the assurance of KPIs consistent evolution, we have focused on ensuring KPIs' consistency along evolution changes by establishing consistency conditions and further consequent propagated changes.

In this case study we consider a KPI system formed by 11 indicators obtained from the paper [DiPS16], where authors use this system to illustrate their framework for collaborative construction and maintenance of dictionaries of performance indicators (which, at the same time, is based on the KPIOnto ontology).

The KPI System formed by 11 KPIs (6 atomic and 5 compound) linked by the following formulas (the ID is of our own) is the following:

1. **NH**- NumHours
2. **HC**-HourlyCost
3. **O**- Overhead
4. **TC**- TravelCosts
5. **PTT**- PersonnelTrainingTime
6. **HR**- HourRate
7. **PC**- PersonnelCosts=NumHours*HourlyCost*(Overhead+1) = **NH*HC*(O+1)**
8. **C**- Costs=TravelCosts+PersonnelCosts = **TC+PC**
9. **PTC**- PersonnelTrainingCosts=HourlyCost*PersonnelTrainingTime = **HC*PTT**
10. **TeC**- TeachCosts=PersonnelTrainingTime*HourRate= **PTT*HR**
11. **IED**- InvestmentInEmplDevelopment=PersonnelTrainingCosts+TeachCost = **PTC+TeC**

As advanced previously, we have used this case study to show the interest in ensuring evolution changes' consistency by establishing consistency conditions and further consequent propagated changes (*consistent evolution*). More specifically, we have considered several evolution changes to be applied to the KPI system presented in this case study so that they provoked different incoherence situations in the system, some of them requiring consequent propagated changes to be performed to ensure the final system's consistency. These evolution changes have shown the importance of establishing well-defined consistency properties and required consequent propagation patterns to guarantee the KPI system's consistency.

KPI System in Formula

```
KPIBasic1 is KPI(0,"KPIBasic1",Active,0)
KPIFormula(0,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(0,"", "",Atomic),KPIBasic1)
basicValue(KPIBasic1,1)
//NumHours
KPINH is KPI(1, "KPINH",Active,0)
KPIFormula(1,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(1,"", "",Atomic),KPINH)
basicValue(KPINH,2)
```

```

//HourlyCost
KPIHC is KPI(2,"KPIHC",Active,0)
KPIFormula(2,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(2,"", "",Atomic),KPIHC)
basicValue(KPIHC,4)
//Overhead
KPIO is KPI(3,"KPIO",Active,0)
KPIFormula(3,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(3,"", "",Atomic),KPIO)
basicValue(KPIO,5)

//TravelCosts
KPITC is KPI(4,"KPITC",Active,0)
KPIFormula(4,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(4,"", "",Atomic),KPITC)
basicValue(KPITC,2)
//PersonnelTrainingTime
KPIPTT is KPI(5,"KPIPTT",Active,0)
KPIFormula(5,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(5,"", "",Atomic),KPIPTT)
basicValue(KPIPTT,2)
//HourRate
KPIHR is KPI(6,"KPIHR",Active,0)
KPIFormula(6,"", "",Atomic)
KPIElement_belongs_to_Version(KPIFormula(6,"", "",Atomic),KPIHR)
basicValue(KPIHR,2)
//PersonnelCosts=NumHours*HourlyCost*(Overhead+1) = NH*HC*(O+1)
//KPIPCa=O+1
KPIPCa is KPI(71,"KPIPCa",Active,0)
KPIFormula(71,"KPIO+KPIBasic1", "+",Compound)
KPIElement_belongs_to_Version(KPIFormula(71,"KPIO+KPIBasic1", "+",Compound),KPIPCa)
is_based_on(KPIPCa,KPIBasic1)
is_based_on(KPIPCa,KPIO)
//KPIPCb=KPIPCa*KPIHC
KPIPCb is KPI(72,"KPIPCb",Active,0)
KPIFormula(72,"KPIPCa*KPIHC", "*",Compound)
KPIElement_belongs_to_Version(KPIFormula(72,"KPIPCa*KPIHC", "*",Compound),KPIPCb)
is_based_on(KPIPCb,KPIPCa)
is_based_on(KPIPCb,KPIHC)
KPIPC is KPI(7,"KPIPC",Active,0)
KPIFormula(7,"KPINH*KPIPCb", "*",Compound)
KPIElement_belongs_to_Version(KPIFormula(7,"KPINH*KPIPCb", "*",Compound),KPIPC)
is_based_on(KPIPC,KPINH)
is_based_on(KPIPC,KPIPCb)
//C- Costs=TravelCosts+PersonnelCosts = KPITC*KPIPC
KPIC is KPI(8,"KPIC",Active,0)
KPIFormula(8,"KPITC*KPIPC", "*",Compound)
KPIElement_belongs_to_Version(KPIFormula(8,"KPITC*KPIPC", "*",Compound),KPIC)
is_based_on(KPIC,KPITC)
is_based_on(KPIC,KPIPC)
//PTC- PersonnelTrainingCosts=HourlyCost*PersonnelTrainingTime = HC*PTT
KPIPTC is KPI(9,"KPIPTC",Active,0)
KPIFormula(9,"KPIHC*KPIPTT", "*",Compound)
KPIElement_belongs_to_Version(KPIFormula(9,"KPIHC*KPIPTT", "*",Compound),KPIPTC)
is_based_on(KPIPTC,KPIHC)
is_based_on(KPIPTC,KPIPTT)
//TeC- TeachCosts=PersonnelTrainingTime*HourRate= PTT*HR
KPITeC is KPI(10,"KPITeC",Active,0)
KPIFormula(10,"KPIPTT*KPIHR", "*",Compound)
KPIElement_belongs_to_Version(KPIFormula(10,"KPIPTT*KPIHR", "*",Compound),KPITeC)
is_based_on(KPITeC,KPIPTT)
is_based_on(KPITeC,KPIHR)
//IED- InvestmentInEmplDevelopment=PersonnelTrainingCosts+TeachCost = PTC+TC
KPIIED is KPI(11,"KPIIED",Active,0)
KPIFormula(11,"KPIPTC+KPITeC", "+",Compound)
KPIElement_belongs_to_Version(KPIFormula(11,"KPIPTC+KPITeC", "+",Compound),KPIIED)
is_based_on(KPIIED,KPIPTC)
is_based_on(KPIIED,KPITeC)

```

Experiment 1. Modifying a KPI calculation rule so that it causes circular dependencies

Violated constraint: coherence

Context: Given $PTC = HC * PTT$, we update PTC calculation rule so that it depends on IED, for example: $PTC = IED * PTT$

Formula instructions using our transformations, choosing all customizable options regarding traceability aspects:

```
load path\AFFECTK\FormulaPrototype\FormulaPrototype.4ml
transform UpdateCalculationRule < IN.KPI(9,"KPIPTC",Active,0), -1, "KPIIED*KPIPTT",
"*",IN.KPI(11,"KPIIED",Active,0), IN.KPI(5,"KPIPTT",Active,0),true,"motivation change", true,true> KPISystem
```

Formula results (we only include the query results):

Considering that:

UpdateCalculationRule.conformsTransform := !nonUniqueness & !nonActiveState & !incoherence.

Please note the **true** evaluation result of **UpdateCalculationRule.incoherence** and, thus, the **false** value of the **UpdateCalculationRule.conformsTransform** query.

```
Query evaluation results:
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
IN.KPIMetamodel.nonUniqueness => False
UpdateCalculationRule.@user_conforms => False
UpdateCalculationRule.incoherence => True
UpdateCalculationRule.conformsTransform => False
UpdateCalculationRule.nonActiveState => False
UpdateCalculationRule.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => False
out.KPIEvolutionMetamodel.conforms => False
out.KPIMetamodel.@user_conforms => False
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.incoherence => True
out.KPIMetamodel.conforms => False
out.KPIMetamodel.granularity => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
out.KPIMetamodel.nonUniqueness => False
```

Experiment 2. Deletion of a KPI that has others that depend on it

Violated constraint: coherence

Context: Deletion of KPI TC- TravelCosts which is included in the formula of the indicator:

$$C- \text{ Costs} = \text{TravelCosts} + \text{PersonnelCosts} = \text{TC} + \text{PC}$$

Formula instructions using our transformations:

```
load path to FormulaPrototype.4ml  
transform DeleteKPI<IN.KPI(4,"KPITC",Active,0), true, " motivation change", true, true> KPISystem
```

Formula results (we only include the query results):

Considering that:

DeleteKPI.conformsTransform := !nonActiveState & !incoherence.

Please note the **true** evaluation result of **DeleteKPI.incoherence** and, thus, the **false** value of the **DeleteKPI.conformsTransform** query.

Query evaluation results:

```
DeleteKPI.@user_conforms => False  
DeleteKPI.incoherence => True  
DeleteKPI.conformsTransform => False  
DeleteKPI.nonActiveState => False  
IN.KPIEvolutionMetamodel.base_conforms => True  
IN.KPIEvolutionMetamodel.conforms => True  
IN.KPIMetamodel.@user_conforms => True  
IN.KPIMetamodel.base_conforms => True  
IN.KPIMetamodel.incoherence => False  
IN.KPIMetamodel.conforms => True  
IN.KPIMetamodel.granularity => False  
IN.KPIMetamodel.repeatedName => False  
IN.KPIMetamodel.targetExpectations => False  
IN.KPIMetamodel.nonUniqueness => False  
out.KPIEvolutionMetamodel.base_conforms => True  
out.KPIEvolutionMetamodel.conforms => True  
out.KPIMetamodel.@user_conforms => True  
out.KPIMetamodel.base_conforms => True  
out.KPIMetamodel.incoherence => False  
out.KPIMetamodel.conforms => True  
out.KPIMetamodel.granularity => False  
out.KPIMetamodel.repeatedName => False  
out.KPIMetamodel.targetExpectations => False  
out.KPIMetamodel.nonUniqueness => False
```

Experiment 3. Introduction of a new PI with the same name as another one

Violated constraint: nonRepeatedName

Context: We create a new indicator identified as **C** (we repeat the name) with the teach and personnel costs:

$$\mathbf{C- Costs = TeachCosts + PersonnelCosts = TeC + PC}$$

Formula instructions using our transformations:

Since indicators do not have target, we suppose, as example, that the new KPI must be "lessThan" 78.

```
load path to FormulaPrototype.4ml

transform CreateKPI <"KPIC", 29, Compound, "KPITeC + KPIPC", lessThan, 78, -1,
"+", IN.KPI(10, "KPITeC", Active, 0), IN.KPI(7, "KPIPC", Active, 0), true, "new goals" > KPISystem
```

Formula results (we only include the query results):

Considering that:

CreateKPI.conformsTransform := !incoherence & !repeatedName & !nonUniqueness.

Please note the **true** evaluation result of **CreateKPI.repeatedName** and, thus, the **false** value of the **CreateKPI.conformsTransform** query.

```
Query evaluation results:

CreateKPI.@user_conforms => False
CreateKPI.incoherence => False
CreateKPI.conformsTransform => False
CreateKPI.repeatedName => True
CreateKPI.nonUniqueness => False
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
IN.KPIMetamodel.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.conforms => False
out.KPIMetamodel.granularity => False
out.KPIMetamodel.repeatedName => True
out.KPIMetamodel.targetExpectations => False
out.KPIMetamodel.nonUniqueness => False
```

Experiment 4. Introduction of a new PI with the formula than another one

Violated constraint: uniqueness

Context: We create a new indicator with the same formula as TeC:

TeC- TeachCosts=PersonnelTrainingTime*HourRate= PTT*HR

Formula instructions using our transformations:

```
load path to FormulaPrototype.4ml

transform CreateKPI <"KPINew", 27, Compound, "KPIPTT*KPIHR", lessThan, 78, -1, "+",
IN.KPI(5,"KPIPTT",Active,0), IN.KPI(6,"KPIHR",Active,0),true, "new goals" > KPISystem
```

Formula results (we only include the query results):

Considering that:

CreateKPI.conformsTransform := !incoherence & !repeatedName & !nonUniqueness.

Please note the **true** evaluation result of **CreateKPI.nonUniqueness** and, thus, the **false** value of the **CreateKPI.conformsTransform** query.

```
Query evaluation results:

CreateKPI.@user_conforms => False
CreateKPI.incoherence => False
CreateKPI.conformsTransform => False
CreateKPI.repeatedName => False
CreateKPI.nonUniqueness => True
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
IN.KPIMetamodel.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => False
out.KPIEvolutionMetamodel.conforms => False
out.KPIMetamodel.@user_conforms => False
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.conforms => False
out.KPIMetamodel.granularity => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
out.KPIMetamodel.nonUniqueness => True
```

Case study CS4

Reference: M. De Sanctis, L. Iovino, M. Rossi M, M. Wimmer. MIKADO: a smart city KPIs assessment modeling framework. *Software and Systems Modeling* 2021; 21: 281–309.

Evaluated characteristic: *functional completeness*, which is related to the capability of a product to provide a set of functions that covers all the specified tasks and intended users' objectives [SQuaRE].

Experiments' goal. Evaluating the functional completeness of our tool.

In this case study we use a real-world KPI system proposed for evaluating the degree of smartness and sustainability of smart cities.

As presented in the paper, the changes that can be performed on the smart cities models can be grouped into three categories:

- *Additive changes* adding new modeling concepts related to smart cities does not impact existing KPIs definition models. New data providers can be defined in the smart city model, with the effect that they will be available in the KPIs definition as parameters. New stakeholders as well as new data points may be defined which subsequently will be referable by the shown tracing mechanism.
- *Subtractive changes*. Removing concepts from the smart city model may impact the KPIs definition models. For instance, removing a data provider affects the KPI definition if a parameter is referring to the removed data package.
- *Structural changes*. A typical example is when the name of the parameter used in the KPIs definition model is changed in the smart city model.

Authors state that also KPIs evolution can essentially be of the same categories as explained before for the smart city models whose implementation can be realized via a few modeling steps on the KPIs definition model:

- *Additive changes*: can be motivated by the introduction of new services on a city, giving rise to new KPIs.
- *Subtractive changes*: imply the deletion of obsolete KPIs, when it is no longer interesting to calculate them on a city..
- *Structural changes*: provoke changes in the definition of KPIs (calculation rule, target,...).

In this case study we evaluate whether our proposal provides suitable functions to cover these three categories of changes, by performing the concrete evolution changes presented by the authors in each category.

Before moving on to present the experiments we have carried out, we will now present the different indicators considered in the paper⁶.

⁶ For convenience when referring to the indicators, we have assigned them their own numbering.

Atomic indicators

- The KPI *Air Pollution* (AP) measures the air quality based on the values reported for specific pollutants. It is part of the dimension *Environment*, sub-dimension *Environment* and category *Air Quality*. The *Air Pollution* is based on the *Air Quality Index (AQI)* KPI for specific pollutants, such as particulate matter (PM), nitrogen dioxide (NO₂), sulfur dioxide (SO₂) and ozone (O₃). Usually, the AQI is calculated as in the following formula, where p refers to the pollutant, whereas the legal limit is established by the law⁷:

$$AQI_p = \frac{\text{measured concentration } p}{\text{legal limit}} \times 100$$

Taking this into account, different concrete indicators are considered regarding air pollution:

KPI1- AQI_{PM2.5}: Air quality index for particulate matter, where the legal limit is 25.

KPI2- AQI_{PM10}: Air quality index for particulate matter, where the legal limit is 50.

KPI3- AQI_{NO2}: Air quality index for nitrogen dioxide, where the legal limit is 200.0.

KPI4- AQI_{SO2}: Air quality index for sulfur dioxide, where the legal limit is 350.

KPI5- AQI_{O3}: Air quality index for ozone, where the legal limit is 180.

- The KPI *Green Areas* (GA) measures the green area in the city per 100.000 inhabitants. It belongs to the dimension *Environment*, sub-dimension *Environment* and category *Public Spaces & Nature*. It is calculated as we indicate next, taking in input two parameters, i.e., the total area of green space in the city in hectares and the city's population.

$$KPI7- GA = \frac{\text{TotalGreenArea}}{\frac{1}{100000} \times \text{CityPop}}$$

- The KPI *real-time transport monitoring* (TM) refers to the (un)availability of real-time transport monitoring systems in the candidate city; thus, it can be represented by a Boolean value. It belongs to the dimension *Infrastructure* and sub-dimension *Digital Infrastructure*.

KPI8- TM= (un)availability of real-time transport monitoring systems

- The KPI *Number of mobile applications* (MA) indicates the number of mobile applications available in the city (e.g., food delivery, car sharing), and it is obtained by collecting mobile applications information from different stores. The KPI belongs to the dimension *digital competencies* which is a sub-dimension of *competencies*.

KPI9-MA= number of mobile applications available in the city

- The following two KPIs are structured in themes and sub-themes. They refer to the NO₂ emissions and PM_{2.5} emissions KPIs. Both of them belong to the theme *Planet* and sub-theme *Pollution & Waste* and take as input the emissions of the corresponding pollutant p and the city's population: $\text{Emissions}_p = \frac{p}{\text{CityPop}}$.

Taking this into account, the indicators are:

KPI10- E_{PM2.5}: Particulate matter 2.5 emissions

KPI11- E_{NO2}: Nitrogen dioxide emissions

⁷ <http://apollon-project.it/2019/12/10/indice-di-qualita-dellaria-iqa/>

- The *wireless broadband coverage* KPI is defined as: broadband coverage + 4G coverage

KPI12 = broadband coverage + 4G coverage

Compound indicators

The worst *Air Quality Index* AQI_p determines the *Air Pollution* KPI, as in the following formula:

KPI6-AP = max (AQI_{PM2.5}, AQI_{PM10}, AQI_{NO2}, AQI_{SO2}, AQI_{O3}) = max (KPI1, KPI2, KPI3, KPI4, KPI5)

KPI System in Formula

```
//KPI1- AQIPM2.5
KPI1 is KPI(1, "KPI1", Active, 0)
KPIFormula(1, "(measuredConcentrationp / 25)*100", "*", Atomic)
KPIElement_belongs_to_Version(KPIFormula(1, "(measuredConcentrationp / 25)*100", "*", Atomic),
KPI1)
basicValue(KPI1, 1)
//KPI2- KPI2- AQIPM10
KPI2 is KPI(2, "KPI2", Active, 0)
KPIFormula(2, "(measuredConcentrationp / 50)*100", "*", Atomic)
KPIElement_belongs_to_Version(KPIFormula(2, "(measuredConcentrationp / 50)*100", "*", Atomic),
KPI2)
basicValue(KPI2, 2)
//KPI3- AQINO2
KPI3 is KPI(3, "KPI3", Active, 0)
KPIFormula(3, "(measuredConcentrationp / 200)*100", "*", Atomic)
KPIElement_belongs_to_Version(KPIFormula(3, "(measuredConcentrationp / 200)*100", "*", Atomic),
KPI3)
basicValue(KPI3, 3)
//KPI4- AQISO2
KPI4 is KPI(4, "KPI4", Active, 0)
KPIFormula(4, "(measuredConcentrationp / 350)*100", "*", Atomic)
KPIElement_belongs_to_Version(KPIFormula(4, "(measuredConcentrationp / 350)*100", "*", Atomic),
KPI4)
basicValue(KPI4, 4)
//KPI5- AQIO3
KPI5 is KPI(5, "KPI5", Active, 0)
KPIFormula(5, "(measuredConcentrationp / 180)*100", "*", Atomic)
KPIElement_belongs_to_Version(KPIFormula(1, "(measuredConcentrationp / 180)*100", "*", Atomic),
KPI5)
basicValue(KPI5, 5)

//KPI7- GA
KPI7 is KPI(7, "KPI7", Active, 0)
KPIFormula(7, "TotalGreenArea/((1/100000)*CityPop)", "/", Atomic)
KPIElement_belongs_to_Version( KPIFormula(7, "TotalGreenArea/((1/100000)*CityPop)", "/",
Atomic), KPI7)
basicValue(KPI7, 7)
//KPI8- TM= (un)availability of real-time transport monitoring systems    KPI8 is KPI(8,
"KPI8", Active, 0)
KPIFormula(8, "", "", Atomic)
KPIElement_belongs_to_Version(KPIFormula(8, "", "", Atomic), KPI8)
basicValue(KPI8, 8)
//KPI9-MA= number of mobile applications available in the city
KPI9 is KPI(9, "KPI9", Active, 0)
KPIFormula(9, "", "", Atomic)
KPIElement_belongs_to_Version(KPIFormula(9, "", "", Atomic), KPI9)
basicValue(KPI9, 9)
//KPI10- EPM2.5: Particulate matter 2.5 emissions
KPI10 is KPI(10, "KPI10", Active, 0)
KPIFormula(10, "p/CityPop", "/", Atomic)
KPIElement_belongs_to_Version(KPIFormula(10, "p/CityPop", "/", Atomic), KPI10)
basicValue(KPI10, 10)
//KPI11- ENO2: Nitrogen dioxide emissions
KPI11 is KPI(11, "KPI11", Active, 0)
KPIFormula(11, "p/CityPop", "/", Atomic)
KPIElement_belongs_to_Version(KPIFormula(11, "p/CityPop", "/", Atomic), KPI11)
basicValue(KPI11, 11)
//KPI12 = broadband coverage + 4G coverage
KPI12 is KPI(12, "KPI12", Active, 0)
KPIFormula(12, "broadband coverage + 4G coverage", "+", Atomic)
KPIElement_belongs_to_Version(KPIFormula(12, "broadband coverage + 4G coverage", "+", Atomic),
KPI12)
basicValue(KPI12, 12)
```

```
//KPI6-AP = max (AQIPM2.5, AQIPM10, AQINO2, AQISO2, AQIO3)= max (KPI1, KPI2, KPI3, KPI4, KPI5)
KPI6 is KPI(6,"KPI6",Active,0)
KPIFormula(6,"max","/",Compound)
KPIElement_belongs_to_Version(KPIFormula(6,"max","/",Compound), KPI6)
is_based_on(KPI6,KPI1)
is_based_on(KPI6,KPI2)
is_based_on(KPI6,KPI3)
is_based_on(KPI6,KPI4)
is_based_on(KPI6,KPI5)
```

Starting from the previous indicators, we have performed the corresponding evolution changes stated in the paper, each falling under one of the cited categories.

- *Experiment 1. Additive changes.* Creation of a new KPI.
- *Experiment 2. Subtractive changes.* Deletion of an obsolete KPI.
- *Experiment 3. Structural changes.* Modification of a KPI calculation logic.

Experiment 1. Additive changes. Creation of the indicator BN

Starting from the previous KPIs, a new indicator BN is considered. In particular, the KPI *Bicycle Network* (BN) measures the length of bicycle paths per 100.000 inhabitants. Its calculation is similar to KPI7 except that in this case we take into consideration the total length of bicycle paths in the city. This KPI belongs to the dimension *Economy*, sub-dimension *Infrastructure* and category *Transport*. **KPI13- BN**=
$$\frac{\text{BikePathLength}}{\frac{1}{100000} \times \text{CityPop}}$$

Thus, we have applied the `CreateKPI` transformation, using suitable arguments.

Formula instructions⁸:

```
//load the specification
load path\AFFECTK\FormulaPrototype\FormulaPrototype.4ml
//Invoke the transformation
transform CreateKPI <"KPI13", 13, Atomic, "BikePathLength/((1/100000) * CityPop)", lessThan, 78, -1,
"/", "", "", true, "new needs" > KPISystem
//Show the resulted KPI model
knows CreateKPI#1
```

Formula results:

As a result, Formula has shown that the indicator does not contradict any consistency condition from those implemented by the `CreateKPI` transformation (*nonRepeatedName*, *uniqueness*, and *coherence*). More specifically, the `CreateKPI.conformsTransform` query is evaluated as `true` since queries `CreateKPI.repeatedName`, `CreateKPI.nonUniqueness` and `CreateKPI.incoherence` are evaluated as `false`:

```
Query evaluation results:
CreateKPI.@user_conforms => True
CreateKPI.conformsTransform => True
CreateKPI.incoherence => False
CreateKPI.nonUniqueness => False
CreateKPI.repeatedName => False
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.nonUniqueness => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.nonUniqueness => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
```

⁸ Since target values are not specified, we consider `lessThan 5` for simplicity and `KPITarget` elements are not included in the results.

Additionally, it can be shown that neither the source KPI model (see queries preceded by IN) nor the target KPI model (see queries preceded by out) violate the constraints defined by the Formula domains.

The Formula report also shows the resulting KPI system (including the new KPI in Active state and with 0 as version), together with the corresponding instances related to evolution aspects.

```

1- PACreateKPI13 is PatternApplication("date","User","no comments",CreateKPI,"new needs",Ejecuted)
2- AppliedTraceOperation("",KPI(13,"KPI13",Active,0), PACreateKPI13)
3- AppliedPropagationOperation(
    PACreateKPI13,
    PatternApplication("date","User","no comments", CreateKPI, "InfluencedBy", InStudyOptional),
    "Check whether the application of the 'influences' create pattern to the indicated KPI may imply the
      need for applying the 'influenced by' creation pattern",
    KPI(13,"KPI13",Active,0))
4- AppliedPropagationOperation(
    PACreateKPI13,
    PatternApplication("date","User","no comments", DeleteKPI, "InfluencedBy", InStudyOptional),
    "Check whether the application of the 'influences' create pattern to the indicated KPI may imply the
      need for applying the 'influenced by' delete pattern",
    KPI(13,"KPI13",Active,0))
5- AppliedPropagationOperation(
    PACreateKPI13,
    PatternApplication("date","User","no comments",UpdateCalculationRule,"InfluencedBy",
      InStudyOptional),
    "Check whether the application of the 'influences' create pattern to the indicated KPI may imply the
      need for applying the 'influenced by' update calculation rule pattern",
    KPI(13,"KPI13",Active,0))
6- //KPIs definitions, included the new one
    KPIBN is KPI(13,"KPI13",Active,0)
    KPIFormula(13,"BikePathLength/((1/100000)*CityPop)","/",Atomic)
    KPIElement_belongs_to_Version(KPIFormula(13,"BikePathLength/((1/100000)*CityPop)","/",Atomic),
      KPIBN)
    //Derived calculatedValues, relatedKPIs, KPIValue instances, etc.

```

In this report, we can see that all constraints are satisfied (no `AppliedConsistencyCondition` instances are created with the violated constraints).

Additionally, and according to the CreateKPI pattern, the transformation execution informs the user (by means of suitable instances of `AppliedPropagationOperation`) about the propagation actions she/he could take, choosing between: i) creating other KPIs, considering the creation of the new KPI (line 3), and/or ii) deleting other KPIs, considering the creation of the new KPI (line 4), and/or iii) modifying the calculation rules of other KPIs as appropriate (line 5), making use of the new KPI. In this case, since such propagation actions are not needed to maintain the consistency of the KPI system, they are identified as `InStudyOptional`.

Experiment 2. *Subtractive changes*. Deletion of the indicator $E_{PM2.5}$ (KPI10):

Looking at the KPIs taken into consideration in a work, for the pollutant PM2.5 authors calculate an ad hoc KPI called *PM2.5 emissions* (**KPI10- $E_{PM2.5}$**) and also the air quality index AQI (**KPI1- $AQI_{PM2.5}$**) for the same pollutant. As stated in the paper, this information could be redundant, thus some stakeholder may think of removing the PM2.5 emissions calculation from the KPIs definition model to save space and time for the KPIs assessment. That is

Taking this into account, we have applied the `DeleteKPI` transformation.

Formula instructions:

```
load path to FormulaPrototype.4ml
transform DeleteKPI<IN.KPI(10,"KPI10",Active,0),true,"no longer collected",true,true> KPISystem

//Show the resulted KPI model
knows DeleteKPI#1
```

Formula results:

As a result, Formula has shown that the indicator does not contradict any consistency condition from those implemented by the `DeleteKPI` transformation (*activeState* and *coherence*). More specifically, the `DeleteKPI.conformsTransform` query is evaluated as `true` since queries `DeleteKPI.nonActiveState` and `DeleteKPI.incoherence` are evaluated as `false`:

```
Query evaluation results:

DeleteKPI.@user_conforms => True
DeleteKPI.conformsTransform => True
DeleteKPI.incoherence => False
DeleteKPI.nonActiveState => False
IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.nonUniqueness => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.nonUniqueness => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
```

Additionally, it can be shown that neither the source KPI model (see queries preceded by IN) nor the target KPI model (see queries preceded by out) violate the constraints defined by the Formula domains.

Consequently, the transformation report also suggests (by means of the generated `AppliedPropagationOperation` instances) the person in charge to assess if she/he wants to create or delete other KPIs, considering the deletion of the KPI (lines 3 and 4). These actions are identified as `InStudyOptional`.

Additionally, among the KPIs included in the system it can be observed that the indicator KPI10 appears both with a deprecated version (the previous version 0) and with a deleted one (the new version 1).

```

1- PADeleteKPI10 is PatternApplication("date","User","no comments", DeleteKPI,"no longer
   collected",Ejecuted)
2- AppliedTraceOperation(
   KPI(10,"KPI10",Deprecated,0),
   KPI(10,"KPI10",Deleted,1),
   PADeleteKPI10)
3- AppliedPropagationOperation(
   PADeleteKPI10,
   PatternApplication("date","User","no comments",CreateKPI,"InfluencedBy",InStudyOptional),
   "Check whether the application of the 'influences' delete pattern to the indicated KPI (in the
   Deleted state) may imply the need for applying the 'influenced by' create pattern",
   KPI(10,"KPI10",Deleted,1))
4- AppliedPropagationOperation(
   PADeleteKPI10,
   PatternApplication("date","User","no comments",DeleteKPI,"InfluencedBy",InStudyOptional),
   "Check whether the application of the 'influences' delete pattern to the indicated KPI (in the
   Deleted state) may imply the need for applying the 'influenced by' delete pattern",
   KPI(10,"KPI10",Deleted,1))
5- //KPIs definitions, included the previous and the new version of the indicator
   KPI(10,"KPI10",Deleted,1)
   KPIFormula(10,"p/CityPop","/",Atomic)
   KPIElement_belongs_to_Version(KPIFormula(10,"p/CityPop","/",Atomic),KPI(10,"KPI10",Deleted,1))
   KPI(10,"KPI10",Deprecated,0)
   KPIElement_belongs_to_Version(KPIFormula(10,"p/CityPop","/",Atomic),KPI(10,"KPI10",Deprecated,0))
   //Derived calculatedValues, relatedKPIs, KPIValue instances, etc.

```

Experiment 3. *Structural changes*. Modification of the indicator *wireless broadband coverage* (KPI12):

As stated in the paper, the *wireless broadband coverage* KPI was previously defined as:

KPI12= broadband coverage + 4G coverage.

After the advent of 5G, the calculation for this KPI evolved, by including the 5G network coverage, as:

KPI12 = broadband coverage + 4G coverage + 5G coverage.

That is, the previous value, plus 5G coverage.

To perform this evolution change in the KPI calculation logic of this indicator, we apply the `UpdateCalculationRule` transformation.

Formula instructions:

```
load path\AFFECTK\FormulaPrototype\FormulaPrototype.4ml

transform UpdateCalculationRuleAtomic < IN.KPI(12,"KPI12",Active,0), -1, "broadband coverage + 4G
coverage + 5G coverage", "+", true,"Change KPI12", true, true> KPISystem
//Show the resulted KPI model
knows UpdateCalculationRuleAtomic#1
```

As a result, Formula shows that the consistency conditions are satisfied (we note, for example, query `UpdateCalculationRule.conformsTransform => true`, showing that the queries stated by the transformation are evaluated as `false`, and that the `conforms` queries related to both source and target KPI models are also evaluated to `true`:

```
Query evaluation results:

IN.KPIEvolutionMetamodel.base_conforms => True
IN.KPIEvolutionMetamodel.conforms => True
IN.KPIMetamodel.@user_conforms => True
IN.KPIMetamodel.base_conforms => True
IN.KPIMetamodel.conforms => True
IN.KPIMetamodel.granularity => False
IN.KPIMetamodel.incoherence => False
IN.KPIMetamodel.nonUniqueness => False
IN.KPIMetamodel.repeatedName => False
IN.KPIMetamodel.targetExpectations => False
UpdateCalculationRule.@user_conforms => True
UpdateCalculationRule.conformsTransform => True
UpdateCalculationRule.incoherence => False
UpdateCalculationRule.nonActiveState => False
UpdateCalculationRule.nonUniqueness => False
out.KPIEvolutionMetamodel.base_conforms => True
out.KPIEvolutionMetamodel.conforms => True
out.KPIMetamodel.@user_conforms => True
out.KPIMetamodel.base_conforms => True
out.KPIMetamodel.conforms => True
out.KPIMetamodel.granularity => False
out.KPIMetamodel.incoherence => False
out.KPIMetamodel.nonUniqueness => False
out.KPIMetamodel.repeatedName => False
out.KPIMetamodel.targetExpectations => False
```

In this case, in addition to the reporting information regarding traceability, consistency conditions and propagation aspects, we can also see the evolution of indicator KPI12 as follows:

```

1- PAUpdateKPI12 is PatternApplication("date","User","no comments",UpdateCalculationRule,"change
   KPI12",Ejecuted)
2- AppliedTraceOperation(
   KPIFormula(8,"broadband coverage + 4G coverage","+",Atomic),
   KPIFormula(9,"KPI12+ KPI5GC","+",Atomic),
   PatternApplication("date","User","no comments",
      PAUpdateKPI12)
3- AppliedPropagationOperation(
   PAUpdateKPI12,
   PatternApplication("date","User","no comments",UpdateTarget,"InfluencedBy",InStudyCompulsory),
   "Check whether the application of the 'influences' update calculation rule pattern to the indicated KPI
   may imply the need for applying the 'influenced by' update target pattern",
   KPI(12,"KPI12",Active,1))
4- //KPIs definitions, included the previous and the new version of the indicator DMC
   KPI(12,"KPI12",Deprecated,0)
   KPIFormula(12,"broadband coverage + 4G coverage","+",Atomic)
   KPIElement_belongs_to_Version(KPIFormula(12,"broadband coverage + 4G coverage","+",Atomic),KPI(12,"KPI12",Deprecated,0))
   KPI(12,"KPI12",Active,1)
   KPIFormula(13,"broadband coverage + 4G coverage + 5G coverage","+",Atomic)
   KPIElement_belongs_to_Version(KPIFormula(13,"broadband coverage + 4G coverage + 5G coverage","+",Atomic),KPI(12,"KPI12",Active,1)) ....
//Derived calculatedValues, relatedKPIs relationships and KPIValue instances

```

As it can be seen in line 3, the application of the `UpdateTarget` transformation is indicated as propagated compulsory action in order to check whether the new calculation rule is coherent with the target or status options of the KPI12. As advanced previously, since target aspects are not considered in the definition of the indicators, we have not considered this element of KPIs in our experiments.

We also note that, since there is no other KPIs that include indicator KPI12 in their formulas, the report does not include a `AppliedPropagationOperation` instance to compulsorily propagate a modification of the calculation rule of other indicators.

We also note how both versions of KPI12 are included: the version previous to the evolution change with “Deprecated” state and version identifier 0, and the current one with “Active” state and version identifier 1.

Bibliography

[SQuaRE] ISO/IEC 25010:2023. Systems and software Quality Requirements and Evaluation (SQuaRE)-Product quality model. Standard, International Organization for Standardization; 2023.

[Form] FORMULA- Modeling Foundations. 2020. Website: <http://research.microsoft.com/en-us/projects/formula/>. Last visited on September, 2024.

[SMEP] Reasoning-based prototype. Supplementary material of the AFFECTK Framework. Available at: <https://zenodo.org/record/3627954>. Last visited on September, 2024.