un-crpt-queries_1

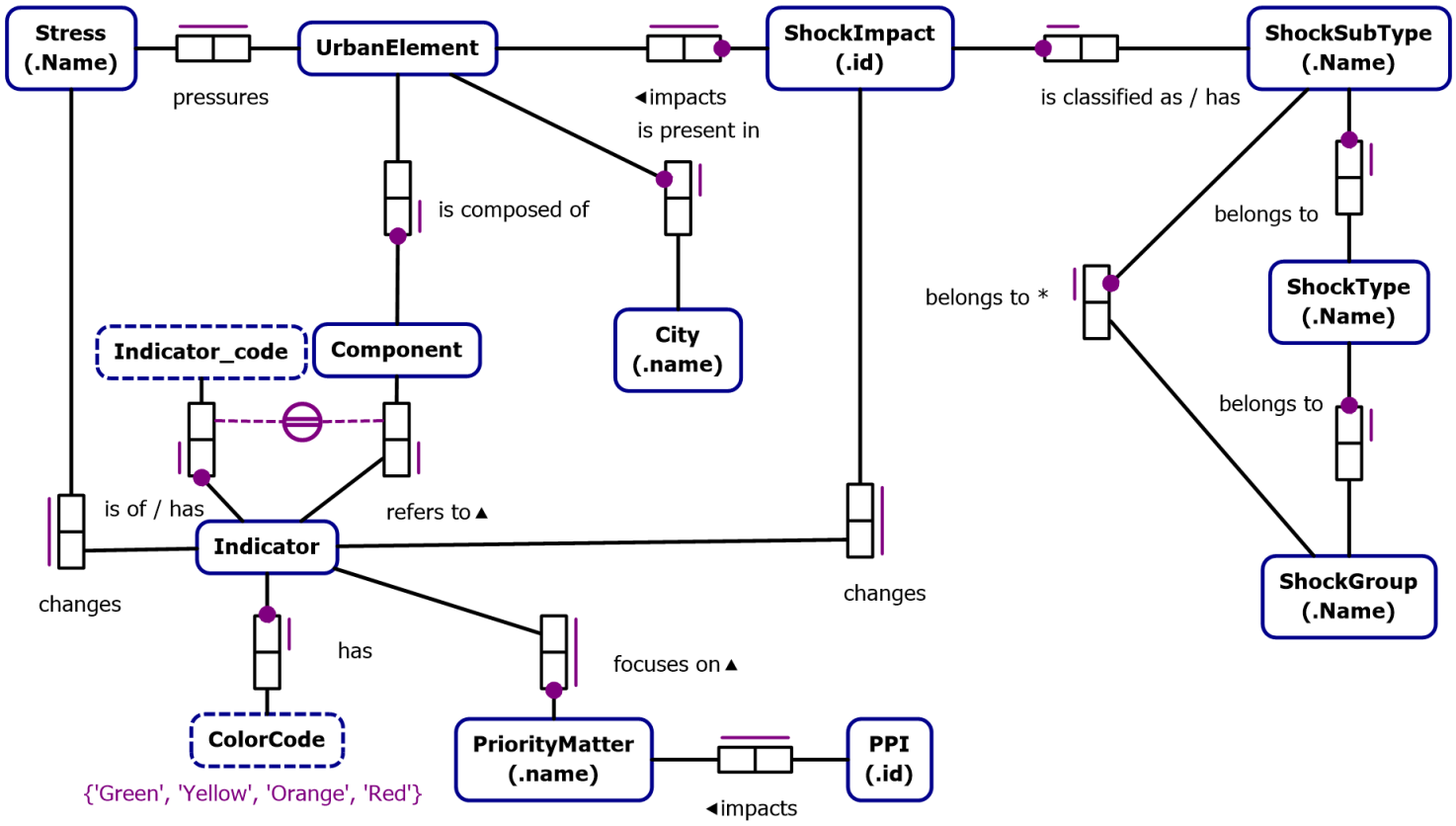
City Resilience Profiling Tool

Types of Questions

... that our tool should be able to answer

- What are the highest risks (i.e. that cause most impact)?
- Which elements and components of the city are affected by floods?
- What are the indicators (i.e. data) relate to the elements affected by floods?
- What are alternatives that would change the risks (i.e. counterfactuals)?
- What is the relationship between shocks (chain of events
- What are the 2nd or 3rd level effects of a flood?

Ontology



Use Case from a real city

- Flooding is a well known problem in the city.
- Around 50% of indicators collected.
- Some data on historical significant shock impacts.
- Interest of the city is economy.
- Following the interest of the city, first assessment would focus on 18 economy indicators and 8 mobility indicators as the base for providing recommendations for action.

Additional analysis using the knowledge graph:

- Impact of flooding showing relevance over the economy of the city
- Impact of shocks triggered by flood
- Impact of all these shocks over all the urban elements

Initial exercise: see the occurrences of flooding over total number of shock impacts

install | un-crpt-queries-shock_type_name_to_shock_in

```
def shock_type_name_to_shock_impact_subtype_name_and_date(stn, sstn, dt) =
  shock_type_name_to_shock_subtype_name(stn, sstn) and
  shock_subtype_name_to_shock_impact(sstn, s) and
  shock_impact_date(s, dt)
  from s
```

query

```
shock_type_name_to_shock_impact_subtype_name_and_date["FLOOD"]
```

query

```
count[shock_type_name_to_shock_impact_subtype_name_and_date["FLOOD"]]
```

For comparison, the total numnber of shocks in the city is shown below

query

```
count[shock_impact_date]
```

Impact of flooding showing relevance over the economy of the city

- Impact on people → Directly affected people
- Direct impact on economy → Loss of working days
- Direct impact on economy → Loss of jobs

install un-crpt-queries-model-3

```
def shock_type_name_to_shock_and_loss_of_working_days(stn, s, lowd) =
  shock_type_name_to_shock_impact(stn, s) and
  shock_impact_loss_of_working_days(s, lowd)
```

query

```
sum[shock_type_name_to_shock_and_loss_of_working_days["FLOOD"]]
```

NOTE: the data may be incomplete, thus the low number of people

install un-crpt-queries-model-4

```
def shock_type_name_to_shock_and_nr_of_people_driectly_affected(stn, s, nr_aff) =
  shock_type_name_to_shock_impact(stn, s) and
  shock_impact_people_directly_affected(s, nr_aff)
```

query

```
sum[shock_type_name_to_shock_and_nr_of_people_driectly_affected["FLOOD"]]
```

install un-crpt-queries-model-12

```
def shock_type_name_to_shock_and_loss_of_jobs(stn, s, l_o_j) =
  shock_type_name_to_shock_impact(stn, s) and
  shock_impact_loss_of_jobs(s, l_o_j)
```

query

```
def output = sum[shock_type_name_to_shock_and_loss_of_jobs["FLOOD"]]
```

install un-crpt-queries-model-14

```
def shock_type_to_indicator(stn, i_c, i_d, i_v) =
  from_shock_type_name(stn, st) and
  shock_subtype_belongs_to_shock_type(sst, st) and
  from_shock_subtype_name(sstn, sst) and
  indicator_code(i, i_c) and
  indicator_value(i, i_v) and
  indicator_description(i, i_d) and
  indicator_to_shock_subtype_name(i, sstn)
  from i, st, sstn, sst
```

query

```
shock_type_to_indicator["FLOOD"]
```

We can now investigate if improving any of these indicators would also help resilience against different type of shocks. This would be challenging using classic methods, e.g. Excel.

install un-crpt-queries-model-15

```
def indicator_code_to_indicator_attr_shock_subtype_name(i_c, i_d, sstn) =
  indicator_code(i, i_c) and
  indicator_description(i, i_d) and
  indicator_to_shock_subtype_name(i, sstn)
  from i
```

query

```
indicator_code_to_indicator_attr_shock_subtype_name["3.2.3.3.2"]
```

NOTE: the data may be incomplete, thus we see no additional shock subtypes here

Impact of shocks triggered by flood

- Shocks triggered by flood?
- Which are the red indicators related to the shocks triggered by flood?

- How much is "Economic Underperformance" is jeopardized by flooding?

install un-crpt-queries-model-5

```
def shock_type_name_to_triggered_shock_subtype_name(stn, tr_sst_n) =
  shock_type_name_to_shock_impact(stn, s) and
  shock_impact_caused_shock(s, tr_sst_n)
  from s
```

query

```
shock_type_name_to_triggered_shock_subtype_name["FLOOD"]
```

install un-crpt-queries-model-7

```
def priority_matter_to_indicator_code_and_value(pm, i_c, i_d, i_v) =
  indicator_code(i, i_c) and
  indicator_description(i, i_d) and
  indicator_value(i, i_v) and
  indicator_to_prioritymatter_name(i, pm)
  from i
```

query

```
priority_matter_to_indicator_code_and_value["Economic Underperformance"]
```

install un-crpt-queries-model-8

```
// List the indicators with Red value for triggered shocks subtypes for a cause
def triggered_red_indicator_and_shock_subtype(cause, i_c, i_d, i_v, sstn) =
  indicator_value(i, i_v) and
  i_v = "Red" and
  indicator_code(i, i_c) and
  indicator_description(i, i_d) and
  indicator_to_shock_subtype_name(i, sstn) and
  sstn = shock_type_name_to_triggered_shock_subtype_name[cause]
  from i
```

query

```
triggered_red_indicator_and_shock_subtype["FLOOD"]
```

If this indicator is to be improved, the resilience against the following shocks could also improve:

install un-crpt-queries-model-16

```
def indicator_code_to_indicator_desc_shock_subtype_name(i_c, i_d, sstn) =
  indicator_code(i, i_c) and
  indicator_description(i, i_d) and
  indicator_to_shock_subtype_name(i, sstn)
  from i
```

query

```
indicator_code_to_indicator_desc_shock_subtype_name["1.2.2.1"]
```

Impact of shocks triggered by flood

- Which are the urban elements affected by flood
- Which are the red indicators related to the urban elements impacted by flood and triggered shocks

install un-crpt-queries-model-13

```
def shock_type_name_to_affected_urban_element(stn, ue) =
  shock_type_name_to_shock_impact(stn, s) and
  shock_impact_affected_urban_elements(s, ue)
  from s
```

query

```
shock_type_name_to_affected_urban_element["FLOOD"]
```

install un-crpt-queries-model-9

```
def triggered_shock_subtype_affecting_urban_elements(cause, sstn, e) =
  sstn = shock_type_name_to_triggered_shock_subtype_name[cause] and
  shock_subtype_name_to_affected_urban_element(sstn, e)
```

query

```
triggered_shock_subtype_affecting_urban_elements["FLOOD"]

// only affected urban elements
//triggered_shock_subtype_affecting_urban_elements["FLOOD"][_]
```

In general, we can go beyond 2nd order shock triggers; we can explore the transitively triggered (n-order) shocks for each shock subtype.

install un-crpt-queries-model-17

```
def shock_trigger_transitively_other_shock = shock_impact_caused_shock . from_shock_subt
def shock_trigger_transitively_other_shock = c, s:
    shock_trigger_transitively_other_shock(c, m) and
    shock_trigger_transitively_other_shock(m, s)
    from m
```

query

```
// Printing type names to make the results more readable
from_shock_subtype_name . shock_subtype_to_impact . shock_trigger_transitively_other_sho
```

Long-term impacts

- Stresses impacting the same elements or indicators as Flood

install un-crpt-queries-model-10

```
def shock_type_to_urban_element_with_stress(stn, uen, strn) =
  stress_name_to_urban_element_name(strn, uen) and
  uen = shock_type_name_to_affected_urban_element[stn]
```

query

```
shock_type_to_urban_element_with_stress["FLOOD"]
```

install | un-crpt-queries-model-11

```
// Intersection of stresses + indicators and shocks + indicators
def shock_to_stress_indicators_impacted(stn, strn, i_c, i_d, i_v, sstn) =
  indicator_code(i, i_c) and
  indicator_value(i, i_v) and
  indicator_description(i, i_d) and
  indicator_to_shock_subtype_name(i, sstn) and
  shock_type_name_to_shock_subtype_name(stn, sstn) and
  indicator_stress(i, str) and
  stress_from_name(strn, str)
  from i, str
```

query

```
shock_to_stress_indicators_impacted["FLOOD"]
```