



Formation report

Model transformation and search of common patterns

Clément Simon - clement.simon@etu.umontpellier.fr

Tutor : Marianne Huchard
Head teacher : Clémentine Nébut

August 28, 2015

Abstract

This report aims to explain how the tool Translaptic works and how it helps in learning transformation patterns from model transformation examples through graph mining. We will see how to use Translaptic, how to improve it. We will also analyse some tests.

Acknowledgments

I want to thank Marianne Huchard and Clémentine Nebut for proposing me this internship subject. I thank them for their leadership, their many details and their responses to different and varied questions. This internship was a confirmation of my desire to do research and it was really pleasant to work with MAREL team.

I also thanks Adel Ferdjoukh for his help brought in the use of a tool.

Thank to Florian that has tolerated me, for carpooling and the daily coffees. Also thank to him, Julia and Marjorie for proofreading.

Contents

1	Introduction	4
2	Approach overview	5
3	Existing tools	6
3.1	Eclipse Modeling Framework	6
3.2	Triptic	6
3.3	Gaston	6
3.4	Formal Concept Analysis	6
3.5	RCAExplore	6
4	Translaptic	8
4.1	Around Gaston	8
4.2	Meta Model	9
4.2.1	Output	10
4.2.2	Creation of metamodel with Eclipse Modeling Framework	10
4.3	Triptic tool	10
4.4	Json generation	10
4.4.1	With Intent Include (JSONWI)	11
4.4.2	Without Intent Include (JSON)	11
4.5	FormalContext MatchingLinks	12
4.6	RCAExplore tool	12
4.7	Metrics	13
5	Analysis of results	15
6	Conclusion	17
A	SFrag0	20
B	SFrag1	21
C	SFrag2	22
D	TFrag0	23
E	TFrag1	24
F	TFrag2	25
G	JSON generated file	26
H	JSON generated file with intent included	30

I XML file generated by RCAExplore	34
Bibliography	37

Chapter 1

Introduction

Translaptic is a tool of translation around Triptic and RCAExplore tools. It was developed during a summer internship at LIRMM in the context of model transformation and graph mining. Translaptic is in two parts, we have the implementation of a searching patterns method and some translations around the Gaston tool. This report will present you the approach and the existing tools, then we will see the Translaptic tool and how to use it. Finally, analysis of experiences will be explained in details.

Chapter 2

Approach overview

This approach sums the approach described in [Saada et al., 2014] and automatizes the algorithm explained in it. The goal of this approach is to find some patterns from source and target files. Those source and target files are graphs each respecting a meta model which represent the started point of our approach.

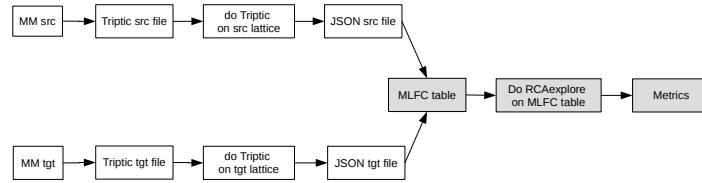


Figure 2.1: Steps for a complete path in Translaptic

In parallel, we can also use GASTON, a subgraph-finding algorithm, to have more informations on the file's graph with those kind of transformations (Fig. 2.2)

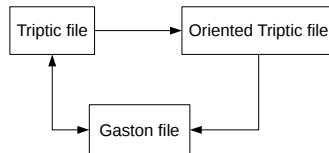


Figure 2.2: Translation make around Triptic and Gaston

Chapter 3

Existing tools

3.1 Eclipse Modeling Framework

The Eclipse Modeling Framework (EMF), [Steinberg et al., 2009], is a modeling framework and code generation facility for building tools integrated in the Eclipse IDE. EMF is a common standard for data models, it includes a metamodel (Ecore) for describing models with default XMI serialization and a reflective API for manipulating EMF objects generically.

3.2 Triptic

GRAAL, from [Liquiere and Sallantin, 1998], construct a Galois lattice for any description language predetermined provided that the two operations of comparison and generalization. Triptic is the tool that implements GRAAL algorithm and give a GUI to manipulate the lattice.

3.3 Gaston

Gaston [Nijssen and Kok, 2004], is a substructure/subgraph-finding algorithm. This tool can read oriented or non-oriented graph. Usually, Gaston is used in searches for substructure that satisfy constraints, such as minimum/maximum frequency, minimum confidence, etc. Gaston exploiting the fact that various substructure are contained in each other. It's a command-line tool.

3.4 Formal Concept Analysis

Formal concept analysis is a mathematical theory of data analysis using formal contexts and concept lattice explained in the book [Ganter and Wille, 1997]. Each concept in the hierarchy represents the set of objects sharing the same values for a certain set of properties; and each sub-concept in the hierarchy contains a subset of the objects in the concepts above it. A concept can be defined by his extent and his intent : the extent is the set of objects which belongs to the concept, the intent is the set of properties shared by those objects.

3.5 RCAExplore

Relational Concept Analysis (RCA) is an extension of the Formal Concept Analysis process (cf. [Ganter and Wille, 1997]) to consider several contexts and relations between them. RCAExplore is an implementation of the RCA process with the possibility to modify the data considered

before each iteration of the process. RCAExplore proposes a "Relational Context Family editor, and interactive Concept Lattice Family Generator and a Concept Lattice Family browser" [Dolques,].

Chapter 4

Translaptic

This tool is a set of functionalities for graph mining. It is composed of multiples steps :

Around Gaston some translators around a Triptic file and a Gaston file ;

Meta Model from a meta model to a file used by Triptic tool ;

Triptic tool the Triptic tool ;

JSON generation from Triptic result to a json file ;

FormalContext MatchingLinks from a JSON file to a matching links between two JSON files ;

RCAExplore tool the RCAExplore tool ;

Metrics from RCAExplore result to a metrics file.

We are going to explain each functionalities one after the other for more clarity.

4.1 Around Gaston

Gaston is a substructure/subgraph-finding algorithm. With Gaston algorithm we can read a graph, find a subgraph and give its number of occurrence. We created a translator between Triptic language and Gaston language.

There is an example of graph (fig. 4.1) and the corresponding files (list. 4.4 and list. 4.2) . Triptic has a system of label with string unlike Gaston that uses numbers. That's why a correspondence file (list. 4.3) is created for each translation, this file allows to translate graphs from Triptic to Gaston and from Gaston return to Triptic with labels conserved. Even if multiple files are translated to Gaston, a unique correspondence file is created with all correspondences of the different files.

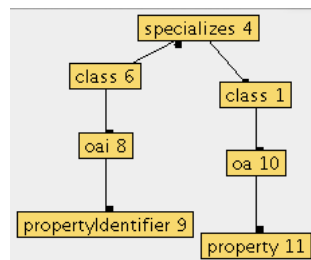


Figure 4.1: Example for translation from Triptic to Gaston

Listing 4.1: Triptic file

```

7
1 1 6 class 110 40 / 3
2 1 1 specializes 60 10 / 4
3 2 2 4 class 10 40 / 3
4 1 5 oai 10 70 / 5
5 0 propertyIdentifier 10 110 / 6
6 1 7 oa 110 70 / 7
7 0 property 110 110 / 8

```

Listing 4.2: Gaston file

```

t #SFrag0
v 0 0
v 1 1
v 2 0
v 3 2
v 4 3
v 5 4
v 6 5
e 0 5 0
e 1 0 1
e 2 3 0
e 2 1 0
e 3 4 2
e 5 6 4

```

Listing 4.3: Correspondence file

{3=propertyIdentifier , 2=oai , 1=specializes , 0=class , 5=property , 4=oa}

Another translator to transform a graph into an explicit oriented graph in Triptic (list. 4.5) has also been created. With this, we can use Gaston on oriented graphs.

Listing 4.4: Triptic file

```

7
1 1 6 class 110 40 / 3
2 1 1 specializes 60 10 / 4
3 2 2 4 class 10 40 / 3
4 1 5 oai 10 70 / 5
5 0 propertyIdentifier 10 110 / 6
6 1 7 oa 110 70 / 7
7 0 property 110 110 / 8

```

Listing 4.5: Triptic oriented file

```

19
1 1 2 class 110 40 / 1
2 2 1 3 classsrc 160 90 / 1
3 2 2 8 classtgt 210 140 / 1
4 1 5 specializes 60 10 / 4
5 2 4 6 specializessrc 110 60 / 4
6 2 5 5 specializestgt 160 110 / 4
7 2 8 10 class 10 40 / 7
8 2 7 9 classsrc 60 90 / 7
9 2 8 16 classtgt 110 140 / 7
10 2 7 11 classsrc 160 190 / 7
11 2 10 10 classtgt 210 240 / 7
12 1 13 oai 10 70 / 12
13 2 12 14 oaisrc 60 120 / 12
14 2 13 15 oaitgt 110 170 / 12
15 0 propertyIdentifier 10 110 / 15
16 1 17 oa 110 70 / 16
17 2 16 18 oasrc 160 120 / 16
18 2 17 19 oatgt 210 170 / 16
19 0 property 110 110 / 19

```

4.2 Meta Model

With Translaptic tool you can create a metamodel in order to create graphs and transform those graphs to a Triptic file (the reverse translation, from a Triptic file to a graph readable by

EMF, has also been created). We will see the output of the translation and how to create a new metamodel, if existing ones do not suit you.

4.2.1 Output

Translaptic will translate your model, created from your metamodel, in a file readable by Triptic tool in order to use Triptic on your graphs.

4.2.2 Creation of metamodel with Eclipse Modeling Framework

To create a new metamodel you need to install dependencies in Eclipse, more precisely Eclipse Modeling Framework and its components. After that you can create an ecore file which is the model of your metamodel. The generator EMF will generate the metamodel corresponding for you and you only need to add it to Translaptic project. To do that you will need to modify the code of Translaptic. First, add your metamodel's keyword in the list of possible transformation in the main class of the mainpackage of Translaptic (list. 4.6).

Listing 4.6: keyword for metamodel in main class

```
case "-MYKEYWORDtoTriptic":
    for (int i=1;i<args.length;i++){
        new metatotriptic.Translate(args[i], "MYKEYWORD");
    }
break;
```

Then, add the keyword in the switch function of the Translate class of metatotriptic package, in method: LoadEMFModel(String filePath, String MMPackage) (list. 4.7). The role's keyword is the detection of which instance of metamodel that will be loaded for the concerned file.

Listing 4.7: generate the right xml resource from the right package

```
case "MYKEYWORD":
    xmlMap.setNoNamespacePackage(MyNewMetaModelMMPackage.eINSTANCE);
break;
```

This set of operations will let you use Translaptic with your own metamodel.

4.3 Triptic tool

Triptic tool has been lightly modified to create output file needed by Translaptic. In fact, intents and extents of a lattice are now generated at the same time of the concerned lattice. You can now create a lattice which has a set of graph for intents and it will be connected to files which implements those intents. The output of Triptic is this set of intents files. To run Triptic, a file which list the graphs files (list. 4.8) is created, the graphs files are the same as previously (list. A to list. C).

Listing 4.8: Lattice file

```
SFrag0
SFrag1
SFrag2
```

4.4 Json generation

After the generation of lattice by Triptic, Translaptic called a translator for JSON. We choosed JSON because it's a lightweight data-interchange format. Easily readable and writable, JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

This translator will read the lattice file and for each intents it found, it will create an object in a list. We have two possible implementations for the JSON file.

4.4.1 With Intent Include (JSONWI)

This implementation will generate a JSON file where the intents of the lattice will be included in it. For example, if we take the first graph example (fig. A) and some other (fig. B and fig. C), the JSON with intents created will be at list. 4.9.

Listing 4.9: JSONWI file

```
{
  "Lattice" : {
    "name" : "SourceFragmentsSave",
    "nbObjects" : 2,
    "nbConcepts" : 5,
    "objects" : ["[ 1 ]", "[ 2 ]"],
    "concepts" : {
      [...]
      "1" : {
        "id" : "C1",
        "extent" : ["[ 1 ]"],
        "intent" : {
          "name" : "[ 1 ]",
          "nbNodes" : 23,
          "nbEdges" : 22,
          "nodes" : {
            "1" : {
              "label" : "class",
              "linkFrom" : [4],
              "linkTo" : [2]
            },
            [...]
          }
        }
      }
    }
  }
}
```

4.4.2 Without Intent Include (JSON)

The second generation of the JSON file will be the one where every intents has an identify keyword. This keyword is found in the second part of the JSON file, the graphs part ; the first part will always contains the lattice. This generation can be found at list. 4.10.

Listing 4.10: JSON file

```
{
  "Lattice" : {
    "name" : "SourceFragmentsSave",
    "nbObjects" : 2,
    "nbConcepts" : 5,
    "objects" : ["[ 1 ]", "[ 2 ]"],
    "concepts" : {
```

```

[... ]
    "1" : {
        "id" : "C1",
        "extent" : "[ 1 ]",
        "intent" : "[ 1 ]",
        "upperCover" : ["C3"]
    },
    [...]
"Graphs" : {
    "[ 1 ]" : {
        "name" : "[ 1 ]",
        "nbNodes" : 23,
        "nbEdges" : 22,
        "nodes" : {
            "1" : {
                "label" : "class",
                "linkFrom" : [4],
                "linkTo" : [2]
            },
            [...]

```

This two examples of JSON generation can be found complete in appendix chapter (list. H and list. G).

4.5 FormalContext MatchingLinks

The Formal Context Matching Links is a set of links, here between two JSON files. We browsed the two JSON files and we created a link between an extent and his intent. We build a table from the generated links and we can read the extents (ordinate) in common in the differents intents (abscissa) (list. 4.11).

Listing 4.11: FormalContext MatchingLinks

FormalContext MatchingLinks

	sfc_2	sfc_0_1_2	tfc_2	tfc_1_2	tfc_0_1_2	sfc_0_1	tfc_0_1	sfc_1	tfc_1
[2]	x	x	x	x	x				
[0]		x			x	x	x		
[1]		x		x	x	x	x	x	x

4.6 RCAExplore tool

The use of RCAExplore in this case is as a builder of an XML graph which regroup intents and extents of a lattice with a built concept. The RCAExplore tool is more powerful than we use it, but to continue the tracking of this experience we don't need to use all the possibilities of RCAExplore.

The generated XML file (list. 4.12) is structurally close to the previously JSON file. We can find the different concepts and the extents and intents link to it.

Listing 4.12: XML file

```

<RCAExplore_Document>
  <Step nb="0">
    <Lattice numberObj="3" numberAtt="9" numberCpt="6" >
      <Config algo="fca">

```

```

</Config>
<Name>MatchingLinks</Name>
<Object>[ 2 ]</Object>
<Object>[ 0 ]</Object>
<Object>[ 1 ]</Object>
<Attribute>sfc_2</Attribute>
<Attribute>sfc_0_1_2</Attribute>
<Attribute>tfc_2</Attribute>
<Attribute>tfc_1_2</Attribute>
<Attribute>tfc_0_1_2</Attribute>
<Attribute>sfc_0_1</Attribute>
<Attribute>tfc_0_1</Attribute>
<Attribute>sfc_1</Attribute>
<Attribute>tfc_1</Attribute>
<Concept>
  <ID>Concept_MatchingLinks_0</ID>
  <Extent>
    <Object_Ref>[ 2 ]</Object_Ref>
    <Object_Ref>[ 0 ]</Object_Ref>
    <Object_Ref>[ 1 ]</Object_Ref>
  </Extent>
  <Intent>
    <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
    <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
  </Intent>
  <UpperCovers>
  </UpperCovers>
</Concept>
<Concept>
  <ID>Concept_MatchingLinks_1</ID>
  [...]

```

The complete file is in appendix (list. I).

4.7 Metrics

From the XML file generated by RCAExplore we calculate some metrics in browsing this file. There are four metrics already calculated :

Tp_1 couples (src,tgt) in the same concept where src and tgt are not build graphs and do not exist in super concept ;

Tp_n couples (src,tgt) in the same concept where src and tgt are build graphs and do not exist in super concept ;

Tp_n_part_s a part of src graph becomes tgt graph (with (Concept of src) < (Concept of tgt)) ;

Tp_n_part_t the src graph becomes a part of tgt graph (with (Concept of src) > (Concept of tgt)).

The result of the metrics calculator is a table (list. 4.13).

Listing 4.13: Result of metrics

{TP_n=2, TP_1=2, TP_n_part_s=2, TP_n_part_t=1}

To check those metrics, we can generate the lattice graph from RCAExplore output files with the command line `dot -Tpdf step1-0.dot -o output.pdf`

Chapter 5

Analysis of results

With sets of source graphs (list. A to list. C) and target graphs (list. D to list. F) we found couple of source graphs fragment and target graphs fragment in common. Graphs are represented from fig. 5.1 to fig. 5.6.

The metrics obtained for those graphs are in listing 4.13. Those metrics allow to see common parts between source and target lattices. If common parts exist, then we can see repetitions of parts of graphs, patterns transformations are deductible from those repetitions. In contrast, if it does not exist common parts: there are no existing links between models from the two metamodels studied.

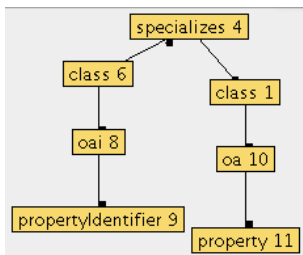


Figure 5.1: Source Graph 0

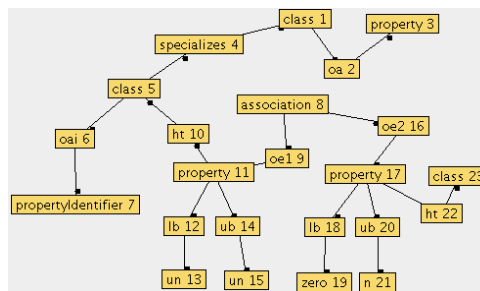


Figure 5.2: Source Graph 1

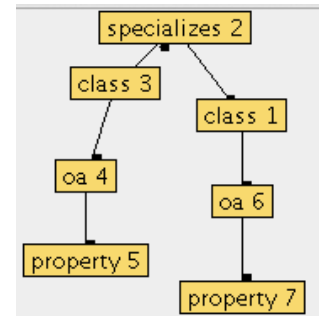


Figure 5.3: Source Graph 2

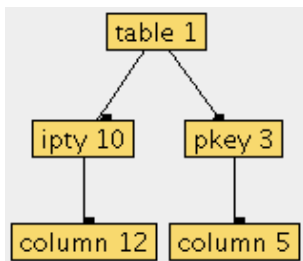


Figure 5.4: Target Graph 0

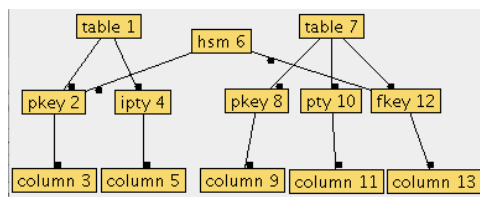


Figure 5.5: Target Graph 1

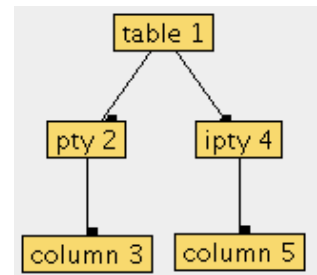


Figure 5.6: Target Graph 2

Chapter 6

Conclusion

To conclude, we have now an automatization of the method explained in [Saada et al., 2014]. The Translaptic tool has also a set of translations, not used in the method, in order to further studies on graphs, especially thanks to GASTON tool. Translaptic is not a complete tool, it is improvable especially compared to the diversity of existing metamodels. To reinforce the list of metamodels in Translaptic, for example a tool that can create metamodels could be used.

List of Figures

2.1	Steps for a complete path in Translaptic	5
2.2	Translation make around Triptic and Gaston	5
4.1	Example for translation from Triptic to Gaston	8
5.1	Source Graph 0	15
5.2	Source Graph 1	15
5.3	Source Graph 2	15
5.4	Target Graph 0	16
5.5	Target Graph 1	16
5.6	Target Graph 2	16

Listings

4.1	Triptic file	9
4.2	Gaston file	9
4.3	Correspondence file	9
4.4	Triptic file	9
4.5	Triptic oriented file	9
4.6	keyword for metamodel in main class	10
4.7	generate the right xml ressource from the right package	10
4.8	Lattice file	10
4.9	JSONWI file	11
4.10	JSON file	11
4.11	FormalContext MatchingLinks	12
4.12	XML file	12
4.13	Result of metrics	13

Appendix A

SFrag0

```
7
1 1 6 class 110 40 / 3
2 1 1 specializes 60 10 / 4
3 2 2 4 class 10 40 / 3
4 1 5 oai 10 70 / 5
5 0 propertyIdentifier 10 110 / 6
6 1 7 oa 110 70 / 7
7 0 property 110 110 / 8
```

Appendix B

SFrag1

```
23
1 1 2 class 280 10 / 3
2 1 3 oa 340 10 / 4
3 0 property 420 10 / 3
4 1 1 specializes 210 50 / 5
5 2 4 6 class 160 100 / 6
6 1 7 oai 210 150 / 7
7 0 propertyIdentifier 70 200 / 8
8 2 9 16 association 280 120 / 7
9 1 11 oe1 280 150 / 7
10 1 5 ht 210 120 / 2
11 3 10 12 14 property 270 160 / 8
12 1 13 lb 220 200 / 2
13 0 un 220 250 / 2
14 1 15 ub 280 200 / 3
15 0 un 280 250 / 2
16 1 17 oe2 300 150 / 3
17 3 18 20 22 property 340 180 / 3
18 1 19 lb 330 250 / 7
19 0 zero 330 280 / 8
20 1 21 ub 370 220 / 7
21 0 n 370 280 / 8
22 1 23 ht 350 150 / 7
23 0 class 360 100 / 8
```

Appendix C

SFrag2

```
7
1 1 6 class 110 40 / 3
2 1 1 specializes 60 10 / 4
3 2 2 4 class 10 40 / 3
4 1 5 oa 10 70 / 5
5 0 property 10 110 / 6
6 1 7 oa 110 70 / 7
7 0 property 110 110 / 3
```


Appendix D

TFrag0

```
5
1 2 2 4 table 80 20 / 3
2 1 3 pkey 40 80 / 4
3 0 column 40 140 / 3
4 1 5 ipty 120 80 / 5
5 0 column 120 140 / 6
```

Appendix E

TFrag1

```
13
1 2 2 4 table 80 80 / 3
2 1 3 pkey 20 140 / 4
3 0 column 20 200 / 3
4 1 5 ipty 100 140 / 5
5 0 column 100 200 / 6
6 1 2 hsm 140 20 / 7
7 3 8 10 12 table 260 80 / 8
8 1 9 pkey 200 140 / 9
9 0 column 200 200 / 10
10 1 11 pty 260 140 / 11
11 0 column 260 200 / 12
12 2 6 13 fkey 320 140 / 13
13 0 column 320 200 / 1
```

Appendix F

TFrag2

```
5
1 2 2 4 table 80 20 / 3
2 1 3 pty 40 80 / 4
3 0 column 40 140 / 3
4 1 5 ipty 120 80 / 5
5 0 column 120 140 / 6
```

Appendix G

JSON generated file

```
{
  "Lattice" : {
    "name" : "SourceFragmentsSave",
    "nbObjects" : 2,
    "nbConcepts" : 5,
    "objects" : [ "[ 1 ]", "[ 2 ]" ],
    "concepts" : {
      "0" : {
        "id" : "C0",
        "extent" : [ ],
        "intent" : "rien",
        "upperCover" : [ "C1", "C2" ]
      },
      "1" : {
        "id" : "C1",
        "extent" : [ "[ 1 ]" ],
        "intent" : "[ 1 ]",
        "upperCover" : [ "C3" ]
      },
      "2" : {
        "id" : "C2",
        "extent" : [ "[ 2 ]" ],
        "intent" : "[ 2 ]",
        "upperCover" : [ "C4" ]
      },
      "3" : {
        "id" : "C3",
        "extent" : [ "[ 0 ]", "[ 1 ]" ],
        "intent" : "[ 0 , 1 ]",
        "upperCover" : [ "C4" ]
      },
      "4" : {
        "id" : "C4",
        "extent" : [ "[ 0 ]", "[ 1 ]", "[ 2 ]" ],
        "intent" : "[ 0 , 1 , 2 ]",
        "upperCover" : [ ]
      }
    }
  },
  "Graphs" : {
    "[ 1 ]" : {
      "name" : "[ 1 ]",
      "nbNodes" : 23,
      "nbEdges" : 22,
      "nodes" : {
        "1" : {
          "label" : "class",
          "linkFrom" : [4],
          "linkTo" : [2]
        },
        "2" : {
          "label" : "oa",
          "linkFrom" : [1],
          "linkTo" : [3]
        },
        "3" : {
          "label" : "property",
          "linkFrom" : [2],
          "linkTo" : [ ]
        },
        "4" : {
          "label" : "specializes",
          "linkFrom" : [5],
          "linkTo" : [1]
        },
        "5" : {
          "label" : "class",
          "linkFrom" : [10],
          "linkTo" : [4,6]
        },
        "6" : {
          "label" : "oai",
          "linkFrom" : [5],
          "linkTo" : [7]
        }
      }
    }
  }
}
```

```

    },
    "7" : {
      "label" : "propertyIdentifier",
      "linkFrom" : [6],
      "linkTo" : []
    },
    "8" : {
      "label" : "association",
      "linkFrom" : [],
      "linkTo" : [9,16]
    },
    "9" : {
      "label" : "oe1",
      "linkFrom" : [8],
      "linkTo" : [11]
    },
    "10" : {
      "label" : "ht",
      "linkFrom" : [11],
      "linkTo" : [5]
    },
    "11" : {
      "label" : "property",
      "linkFrom" : [9],
      "linkTo" : [10,12,14]
    },
    "12" : {
      "label" : "lb",
      "linkFrom" : [11],
      "linkTo" : [13]
    },
    "13" : {
      "label" : "un",
      "linkFrom" : [12],
      "linkTo" : []
    },
    "14" : {
      "label" : "ub",
      "linkFrom" : [11],
      "linkTo" : [15]
    },
    "15" : {
      "label" : "un",
      "linkFrom" : [14],
      "linkTo" : []
    },
    "16" : {
      "label" : "oe2",
      "linkFrom" : [8],
      "linkTo" : [17]
    },
    "17" : {
      "label" : "property",
      "linkFrom" : [16],
      "linkTo" : [18,20,22]
    },
    "18" : {
      "label" : "lb",
      "linkFrom" : [17],
      "linkTo" : [19]
    },
    "19" : {
      "label" : "zero",
      "linkFrom" : [18],
      "linkTo" : []
    },
    "20" : {
      "label" : "ub",
      "linkFrom" : [17],
      "linkTo" : [21]
    },
    "21" : {
      "label" : "n",
      "linkFrom" : [20],
      "linkTo" : []
    },
    "22" : {
      "label" : "ht",
      "linkFrom" : [17],
      "linkTo" : [23]
    },
    "23" : {
      "label" : "class",
      "linkFrom" : [22],
      "linkTo" : []
    }
  }
},
"[ 2 ]" : {
  "name" : "[ 2 ]",
  "nbNodes" : 7,
  "nbEdges" : 6,
  "nodes" : {
    "1" : {
      "label" : "class",
      "linkFrom" : [2],
      "linkTo" : [6]
    },
    "2" : {

```

```

        "label" : "specializes",
        "linkFrom" : [3],
        "linkTo" : [1]
    },
    "3" : {
        "label" : "class",
        "linkFrom" : [],
        "linkTo" : [2,4]
    },
    "4" : {
        "label" : "oa",
        "linkFrom" : [3],
        "linkTo" : [5]
    },
    "5" : {
        "label" : "property",
        "linkFrom" : [4],
        "linkTo" : []
    },
    "6" : {
        "label" : "oa",
        "linkFrom" : [1],
        "linkTo" : [7]
    },
    "7" : {
        "label" : "property",
        "linkFrom" : [6],
        "linkTo" : []
    }
}
},
"[ 0 , 1 ]" : {
    "name" : "[ 0 , 1 ]",
    "nbNodes" : 7,
    "nbEdges" : 6,
    "nodes" : {
        "1" : {
            "label" : "class",
            "linkFrom" : [2],
            "linkTo" : [6]
        },
        "2" : {
            "label" : "specializes",
            "linkFrom" : [3],
            "linkTo" : [1]
        },
        "3" : {
            "label" : "class",
            "linkFrom" : [],
            "linkTo" : [2,4]
        },
        "4" : {
            "label" : "oai",
            "linkFrom" : [3],
            "linkTo" : [5]
        },
        "5" : {
            "label" : "propertyIdentifier",
            "linkFrom" : [4],
            "linkTo" : []
        },
        "6" : {
            "label" : "oa",
            "linkFrom" : [1],
            "linkTo" : [7]
        },
        "7" : {
            "label" : "property",
            "linkFrom" : [6],
            "linkTo" : []
        }
    }
}
},
"[ 0 , 1 , 2 ]" : {
    "name" : "[ 0 , 1 , 2 ]",
    "nbNodes" : 5,
    "nbEdges" : 4,
    "nodes" : {
        "1" : {
            "label" : "class",
            "linkFrom" : [2],
            "linkTo" : [4]
        },
        "2" : {
            "label" : "specializes",
            "linkFrom" : [3],
            "linkTo" : [1]
        },
        "3" : {
            "label" : "class",
            "linkFrom" : [],
            "linkTo" : [2]
        },
        "4" : {
            "label" : "oa",
            "linkFrom" : [1],
            "linkTo" : [5]
        },
        "5" : {

```

```
        "label" : "property",
        "linkFrom" : [4],
        "linkTo" : []
      }
    }
  }
}
```

Appendix H

JSON generated file with intent included

```
{
  "Lattice" : {
    "name" : "SourceFragmentsSave",
    "nbObjects" : 2,
    "nbConcepts" : 5,
    "objects" : [ "[ 1 ]", "[ 2 ]" ],
    "concepts" : {
      "0" : {
        "id" : "C0",
        "extent" : [ ],
        "intent" : { },
        "upperCover" : [ "C1", "C2" ]
      },
      "1" : {
        "id" : "C1",
        "extent" : [ "[ 1 ]" ],
        "intent" : {
          "name" : "[ 1 ]",
          "nbNodes" : 23,
          "nbEdges" : 22,
          "nodes" : {
            "1" : {
              "label" : "class",
              "linkFrom" : [4],
              "linkTo" : [2]
            },
            "2" : {
              "label" : "oa",
              "linkFrom" : [1],
              "linkTo" : [3]
            },
            "3" : {
              "label" : "property",
              "linkFrom" : [2],
              "linkTo" : [ ]
            },
            "4" : {
              "label" : "specializes",
              "linkFrom" : [5],
              "linkTo" : [1]
            },
            "5" : {
              "label" : "class",
              "linkFrom" : [10],
              "linkTo" : [4,6]
            },
            "6" : {
              "label" : "oai",
              "linkFrom" : [5],
              "linkTo" : [7]
            },
            "7" : {
              "label" : "propertyIdentifier",
              "linkFrom" : [6],
              "linkTo" : [ ]
            },
            "8" : {
              "label" : "association",
              "linkFrom" : [ ],
              "linkTo" : [9,16]
            },
            "9" : {
              "label" : "oel",
              "linkFrom" : [8],
              "linkTo" : [11]
            },
            "10" : {
              "label" : "ht",
              "linkFrom" : [11],
              "linkTo" : [5]
            }
          }
        }
      }
    }
  }
}
```



```

    },
    "11" : {
      "label" : "property",
      "linkFrom" : [9],
      "linkTo" : [10,12,14]
    },
    "12" : {
      "label" : "lb",
      "linkFrom" : [11],
      "linkTo" : [13]
    },
    "13" : {
      "label" : "un",
      "linkFrom" : [12],
      "linkTo" : []
    },
    "14" : {
      "label" : "ub",
      "linkFrom" : [11],
      "linkTo" : [15]
    },
    "15" : {
      "label" : "un",
      "linkFrom" : [14],
      "linkTo" : []
    },
    "16" : {
      "label" : "oe2",
      "linkFrom" : [8],
      "linkTo" : [17]
    },
    "17" : {
      "label" : "property",
      "linkFrom" : [16],
      "linkTo" : [18,20,22]
    },
    "18" : {
      "label" : "lb",
      "linkFrom" : [17],
      "linkTo" : [19]
    },
    "19" : {
      "label" : "zero",
      "linkFrom" : [18],
      "linkTo" : []
    },
    "20" : {
      "label" : "ub",
      "linkFrom" : [17],
      "linkTo" : [21]
    },
    "21" : {
      "label" : "n",
      "linkFrom" : [20],
      "linkTo" : []
    },
    "22" : {
      "label" : "ht",
      "linkFrom" : [17],
      "linkTo" : [23]
    },
    "23" : {
      "label" : "class",
      "linkFrom" : [22],
      "linkTo" : []
    }
  },
  "upperCover" : ["C3"]
},
"2" : {
  "id" : "C2",
  "extent" : ["[ 2 ]"],
  "intent" : {
    "name" : "[ 2 ]",
    "nbNodes" : 7,
    "nbEdges" : 6,
    "nodes" : {
      "1" : {
        "label" : "class",
        "linkFrom" : [2],
        "linkTo" : [6]
      },
      "2" : {
        "label" : "specializes",
        "linkFrom" : [3],
        "linkTo" : [1]
      },
      "3" : {
        "label" : "class",
        "linkFrom" : [],
        "linkTo" : [2,4]
      },
      "4" : {
        "label" : "oa",
        "linkFrom" : [3],
        "linkTo" : [5]
      },
      "5" : {

```

```

        "label" : "property",
        "linkFrom" : [4],
        "linkTo" : []
    },
    "6" : {
        "label" : "oa",
        "linkFrom" : [1],
        "linkTo" : [7]
    },
    "7" : {
        "label" : "property",
        "linkFrom" : [6],
        "linkTo" : []
    }
    },
    "upperCover" : ["C4"]
},
"3" : {
    "id" : "C3",
    "extent" : ["[ 0 ]", "[ 1 ]"],
    "intent": {
        "name" : "[ 0 , 1 ]",
        "nbNodes" : 7,
        "nbEdges" : 6,
        "nodes" : {
            "1" : {
                "label" : "class",
                "linkFrom" : [2],
                "linkTo" : [6]
            },
            "2" : {
                "label" : "specializes",
                "linkFrom" : [3],
                "linkTo" : [1]
            },
            "3" : {
                "label" : "class",
                "linkFrom" : [],
                "linkTo" : [2,4]
            },
            "4" : {
                "label" : "oai",
                "linkFrom" : [3],
                "linkTo" : [5]
            },
            "5" : {
                "label" : "propertyIdentifier",
                "linkFrom" : [4],
                "linkTo" : []
            },
            "6" : {
                "label" : "oa",
                "linkFrom" : [1],
                "linkTo" : [7]
            },
            "7" : {
                "label" : "property",
                "linkFrom" : [6],
                "linkTo" : []
            }
        }
    },
    "upperCover" : ["C4"]
},
"4" : {
    "id" : "C4",
    "extent" : ["[ 0 ]", "[ 1 ]", "[ 2 ]"],
    "intent": {
        "name" : "[ 0 , 1 , 2 ]",
        "nbNodes" : 5,
        "nbEdges" : 4,
        "nodes" : {
            "1" : {
                "label" : "class",
                "linkFrom" : [2],
                "linkTo" : [4]
            },
            "2" : {
                "label" : "specializes",
                "linkFrom" : [3],
                "linkTo" : [1]
            },
            "3" : {
                "label" : "class",
                "linkFrom" : [],
                "linkTo" : [2]
            },
            "4" : {
                "label" : "oa",
                "linkFrom" : [1],
                "linkTo" : [5]
            },
            "5" : {
                "label" : "property",
                "linkFrom" : [4],
                "linkTo" : []
            }
        }
    }
}

```

```

    },
    "upperCover" : []
  }
}

```

Appendix I

XML file generated by RCAExplore

```
<RCAExplore_Document>
  <Step nb="0">
    <Lattice numberObj="3" numberAtt="9" numberCpt="6" >
      <Config algo="fca">
      </Config>
      <Name>MatchingLinks</Name>
      <Object>[ 2 ]</Object>
      <Object>[ 0 ]</Object>
      <Object>[ 1 ]</Object>
      <Attribute>sfc_2</Attribute>
      <Attribute>sfc_0_1_2</Attribute>
      <Attribute>tfc_2</Attribute>
      <Attribute>tfc_1_2</Attribute>
      <Attribute>tfc_0_1_2</Attribute>
      <Attribute>sfc_0_1</Attribute>
      <Attribute>tfc_0_1</Attribute>
      <Attribute>sfc_1</Attribute>
      <Attribute>tfc_1</Attribute>
      <Concept>
        <ID>Concept_MatchingLinks_0</ID>
        <Extent>
          <Object_Ref>[ 2 ]</Object_Ref>
          <Object_Ref>[ 0 ]</Object_Ref>
          <Object_Ref>[ 1 ]</Object_Ref>
        </Extent>
        <Intent>
          <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        </Intent>
        <UpperCovers>
        </UpperCovers>
      </Concept>
      <Concept>
        <ID>Concept_MatchingLinks_1</ID>
        <Extent>
          <Object_Ref>[ 2 ]</Object_Ref>
        </Extent>
        <Intent>
          <Attribute_Ref>sfc_2</Attribute_Ref>
          <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_2</Attribute_Ref>
          <Attribute_Ref>tfc_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        </Intent>
        <UpperCovers>
          <Concept_Ref>Concept_MatchingLinks_2</Concept_Ref>
        </UpperCovers>
      </Concept>
      <Concept>
        <ID>Concept_MatchingLinks_2</ID>
        <Extent>
          <Object_Ref>[ 2 ]</Object_Ref>
          <Object_Ref>[ 1 ]</Object_Ref>
        </Extent>
        <Intent>
          <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        </Intent>
        <UpperCovers>
          <Concept_Ref>Concept_MatchingLinks_0</Concept_Ref>
        </UpperCovers>
      </Concept>
      <Concept>
        <ID>Concept_MatchingLinks_3</ID>
        <Extent>
        </Extent>
        <Intent>
          <Attribute_Ref>sfc_2</Attribute_Ref>
          <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_2</Attribute_Ref>
          <Attribute_Ref>tfc_1_2</Attribute_Ref>
          <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
          <Attribute_Ref>sfc_0_1</Attribute_Ref>
        </Intent>
      </Concept>
    </Lattice>
  </Step>
</RCAExplore_Document>
```

```

        <Attribute_Ref>tfc_0_1</Attribute_Ref>
        <Attribute_Ref>sfc_1</Attribute_Ref>
        <Attribute_Ref>tfc_1</Attribute_Ref>
    </Intent>
    <UpperCovers>
        <Concept_Ref>Concept_MatchingLinks_4</Concept_Ref>
        <Concept_Ref>Concept_MatchingLinks_1</Concept_Ref>
    </UpperCovers>
</Concept>
<Concept>
    <ID>Concept_MatchingLinks_4</ID>
    <Extent>
        <Object_Ref>[ 1 ]</Object_Ref>
    </Extent>
    <Intent>
        <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>sfc_0_1</Attribute_Ref>
        <Attribute_Ref>tfc_0_1</Attribute_Ref>
        <Attribute_Ref>sfc_1</Attribute_Ref>
        <Attribute_Ref>tfc_1</Attribute_Ref>
    </Intent>
    <UpperCovers>
        <Concept_Ref>Concept_MatchingLinks_5</Concept_Ref>
        <Concept_Ref>Concept_MatchingLinks_2</Concept_Ref>
    </UpperCovers>
</Concept>
<Concept>
    <ID>Concept_MatchingLinks_5</ID>
    <Extent>
        <Object_Ref>[ 0 ]</Object_Ref>
        <Object_Ref>[ 1 ]</Object_Ref>
    </Extent>
    <Intent>
        <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>sfc_0_1</Attribute_Ref>
        <Attribute_Ref>tfc_0_1</Attribute_Ref>
    </Intent>
    <UpperCovers>
        <Concept_Ref>Concept_MatchingLinks_0</Concept_Ref>
    </UpperCovers>
</Concept>
</Lattice>
</Step>
<Step nb="1">
    <Lattice numberObj="3" numberAtt="9" numberCpt="6">
        <Config algo="fca">
            </Config>
            <Name>MatchingLinks</Name>
            <Object>[ 2 ]</Object>
            <Object>[ 0 ]</Object>
            <Object>[ 1 ]</Object>
            <Attribute>sfc_2</Attribute>
            <Attribute>sfc_0_1_2</Attribute>
            <Attribute>tfc_2</Attribute>
            <Attribute>tfc_1_2</Attribute>
            <Attribute>tfc_0_1_2</Attribute>
            <Attribute>sfc_0_1</Attribute>
            <Attribute>tfc_0_1</Attribute>
            <Attribute>sfc_1</Attribute>
            <Attribute>tfc_1</Attribute>
        </Concept>
        <ID>Concept_MatchingLinks_0</ID>
        <Extent>
            <Object_Ref>[ 2 ]</Object_Ref>
            <Object_Ref>[ 0 ]</Object_Ref>
            <Object_Ref>[ 1 ]</Object_Ref>
        </Extent>
        <Intent>
            <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
            <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        </Intent>
        <UpperCovers>
        </UpperCovers>
    </Concept>
    <Concept>
        <ID>Concept_MatchingLinks_1</ID>
        <Extent>
            <Object_Ref>[ 2 ]</Object_Ref>
        </Extent>
        <Intent>
            <Attribute_Ref>sfc_2</Attribute_Ref>
            <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
            <Attribute_Ref>tfc_2</Attribute_Ref>
            <Attribute_Ref>tfc_1_2</Attribute_Ref>
            <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        </Intent>
        <UpperCovers>
            <Concept_Ref>Concept_MatchingLinks_2</Concept_Ref>
        </UpperCovers>
    </Concept>
    <Concept>
        <ID>Concept_MatchingLinks_2</ID>
        <Extent>
            <Object_Ref>[ 2 ]</Object_Ref>
            <Object_Ref>[ 1 ]</Object_Ref>
        </Extent>
    </Concept>

```

```

        <Intent>
            <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
            <Attribute_Ref>tfc_1_2</Attribute_Ref>
            <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        </Intent>
        <UpperCovers>
            <Concept_Ref>Concept_MatchingLinks_0</Concept_Ref>
        </UpperCovers>
    </Concept>
</Concept>
<Concept>
    <ID>Concept_MatchingLinks_3</ID>
    <Extent>
        </Extent>
    <Intent>
        <Attribute_Ref>sfc_2</Attribute_Ref>
        <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_2</Attribute_Ref>
        <Attribute_Ref>tfc_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>sfc_0_1</Attribute_Ref>
        <Attribute_Ref>tfc_0_1</Attribute_Ref>
        <Attribute_Ref>sfc_1</Attribute_Ref>
        <Attribute_Ref>tfc_1</Attribute_Ref>
    </Intent>
    <UpperCovers>
        <Concept_Ref>Concept_MatchingLinks_4</Concept_Ref>
        <Concept_Ref>Concept_MatchingLinks_1</Concept_Ref>
    </UpperCovers>
</Concept>
</Concept>
<Concept>
    <ID>Concept_MatchingLinks_4</ID>
    <Extent>
        <Object_Ref>[ 1 ]</Object_Ref>
    </Extent>
    <Intent>
        <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>sfc_0_1</Attribute_Ref>
        <Attribute_Ref>tfc_0_1</Attribute_Ref>
        <Attribute_Ref>sfc_1</Attribute_Ref>
        <Attribute_Ref>tfc_1</Attribute_Ref>
    </Intent>
    <UpperCovers>
        <Concept_Ref>Concept_MatchingLinks_5</Concept_Ref>
        <Concept_Ref>Concept_MatchingLinks_2</Concept_Ref>
    </UpperCovers>
</Concept>
</Concept>
<Concept>
    <ID>Concept_MatchingLinks_5</ID>
    <Extent>
        <Object_Ref>[ 0 ]</Object_Ref>
        <Object_Ref>[ 1 ]</Object_Ref>
    </Extent>
    <Intent>
        <Attribute_Ref>sfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>tfc_0_1_2</Attribute_Ref>
        <Attribute_Ref>sfc_0_1</Attribute_Ref>
        <Attribute_Ref>tfc_0_1</Attribute_Ref>
    </Intent>
    <UpperCovers>
        <Concept_Ref>Concept_MatchingLinks_0</Concept_Ref>
    </UpperCovers>
</Concept>
</Lattice>
</Step>
</RCAExplore_Document>

```

Bibliography

- [Dolques,] Dolques, X. Rcaexplore.
- [Ganter and Wille, 1997] Ganter, B. and Wille, R. (1997). *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition.
- [Liquiere and Sallantin, 1998] Liquiere, M. and Sallantin, J. (1998). Structural machine learning with galois lattice and graphs. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 305–313, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Nijssen and Kok, 2004] Nijssen, S. and Kok, J. N. (2004). A quickstart in frequent structure mining can make a difference. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 647–652, New York, NY, USA. ACM.
- [Saada et al., 2014] Saada, H., Huchard, M., Liquière, M., and Nebut, C. (2014). Learning Model Transformation Patterns using Graph Generalization. In Karell Bertet, S. R., editor, *CLA 2014: Eleventh International Conference on Concept Lattices and Their Applications*, volume 1252 of *CEUR-Workshop*, pages 11–22, Košice, Slovakia. Institute of Computer Science, Pavol Jozef Šafárik University in Košice, Ondrej Krídlo.
- [Steinberg et al., 2009] Steinberg, D., Budinsky, F., Paternostro, M., and Merks, E. (2009). *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley Professional, 2nd edition.