The Weak Aliens Problem

Chapter 3 Simplicial approach to Structural Semantic Heterogeneity

Jean-Paul Sansonnet — Aug - Oct 2003

"Along the complex yellow brick road, we go
Chasing the wonders of semantics, we flaw "
J-P Starling, "over the rainbow"

▫

1. Simplicial representation of concepts

1.1 Simplicial complexes

- **Introduction to simplicial complexes**

An introduction to Simplicial Complexes (SC) can be found in E. Valencia PhD thesis, Dec 2000.
This thesis with all the documentation is downloadable online at "http://www.limsi.fr/jps/research/weakaliens".
In the following, a practical introduction to SC is given, step by step, along with the examples.

- **Advantages of Simplicial Complexes over graphs of relations**

The main advantage of Simplicial Complexes over graphs is that they are a generalization of graphs: graphs are 1D simplicial complexes.
Consequence they make it possible for us to refine our model by giving more *precision* to the concepts.

Instead as regarding the concepts of a Tbox as mere atomic symbols (graph nodes) we can give some semantic "thickness" by taking into account their insides, that is the expressions that define them within the Tbox.

If we consider the two approaches to Semantic Heterogeneity discussed in chapter 1,
- **Lexical semantic heterogeneity** is best suited to be treated with a relational graph approach (see chapter 2),
- **Structural semantic heterogeneity** is best suited to be treated with a simplicial approach : this is the purpose of this chapter.

- **Purpose of this chapter**

We will consider two kind of Tbox concepts
  - **Domains** (only extensional) are the simplest form of non atomic concepts.
  - **Tagged structures** (both fields and frames) will introduce the notion of concept hierarchy that is difficult to handle.

Moreover, we will take two approaches to the redefinition of concepts
  - **Direct simplicial redefinition** is encountered when the incidence matrix of the source and the destination complexes is not empty,

- **Polygonal path simplicial redefinition** is encountered when the incidence matrix of source and the destination complexes is empty, but the complexes are still connected.

1.2 Simplicial representation

▪ Extensional domains in the WAP representation

In order to illustrate practically the simplicial representation of concepts we will use extensional domains because it's the simplest form of non atomic concepts.

See chapter 1 of WAP, about domains definitions and generic relations between concepts.

Let Dx, Dy ... be extensional domains in the WAP representation

```
Dx ≐ EXT[{1, 2}];
Dy ≐ EXT[{1, 2, 3, 4}];
Dthings ≐ EXT[{frog, apple, watch}];
```

What can we say of the generic relations between elements x_i of Dx and y_i of Dy?

- SYN[x_i, y_i] ?
- ANT[x_i, y_i] ?
- HYPER[x_i, y_i] ?
- HYPO[x_i, y_i] ?
- SAMECAT[x_i, y_i] ?

It is quite artificial to claim that any of these relations are True or False! That's the point: we don't need to deal with "artificially built" relations but we need do deal with the "real stuff" that is the values over which the domains Dx and Dy are built. So instead of finding a path towards the Ground set, following the generic relations, we'll try to find a path supported by the "contents" of the concepts.

When we apply this policy to the level not only of atomic elements but of non atomic entities like domains for example we can see that the sharing of elements between the structures is a way to build relations from the extensional sub parts that are shared.

▪ The relation K

We define a binary relation $\mathbb{V} \xrightarrow{K} \mathbb{D}$ such that a concept (here a domain D_i) is in relation K with value v if the value $v \in D_i$

Note Conventionally, this relation is named K in Simplicial Complexes theory, because is represents a Complex.

For example let be $\mathbb{D} = \{D_x, D_y, D_z\}$ and $\mathbb{V} = \bigcup \mathbb{D} = \{1,2,3,4\}$ such that we have the following table for the relation K

K	1	2	3	4
Dx	0	0	0	1
Dy	1	1	1	0
Dz	0	1	1	0

The relation K can be represented by the following simplicial complex on the left

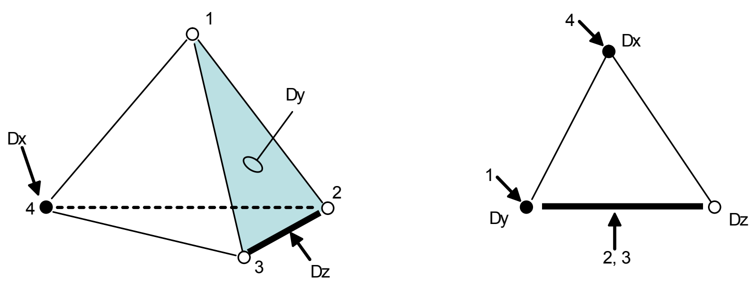

Figure 1. A simplicial representation and its dual form.

On the right we have displayed the dual of the relation К that brings about information on the values, *now considered as concepts* over the domains. We can see that values 1 and 2 cannot be distinguished. This gives us a formal definition for synonymy

Synonymy   we'll say that two concepts are *synonym* if they have the same simplicial representation.

Antonymy   we don't know.

Hyperonymy   is also very simple to define in simplicial terms: HYPER[x, y] will stand iff the simplex of domain y is contained in the simplex of x (and the reverse for HYPO[x,y]).

Note   we'll see that it is the contrary for tagged structures: HYPER[t1,t2] iff t1 has less features than t2 (see Valencia thesis)

Samecat   to do

2. Direct Redefinition of concepts

2.1 Working with domains

■ **Redefinition without polygonal paths**

Direct simplicial redefinition, i.e. without (polygonal) paths, is encountered when the incidence matrix of source and the destination complexes is not empty. It is the simplest situation that can happen, so it will be treated first. Then, a generalization will be developed in the next section, involving polygonal paths.

Now we define new domains Dx, Dy and Dz and we assume that they are black data concepts of a given Tbox. Moreover, we will assume that there exists another agent's Tbox with regard to which a Ground set can be exhibited that contains two domain concepts Dg1 and Dg2. Let be the following К table for all those concepts:

К	1	2	3	4
Dx	0	0	0	1
Dy	1	1	1	0
Dz	0	1	1	0
Dg1	0	0	1	1
Dg2	0	1	1	1

Note   What is the situation of the value "1"? it is not in the ground values but we are not sure that there doesn't exist another domain DX of the other agent that contains 0 and such that DX is not in the Ground set. The consequence of such a situation is that "1" can be known by both agents without being in the extension of the domains that are in the Ground set!

This table can be illustrated by the following figure where

- Dg1 is the edge {3,4} = <<3>, <4>, <3,4>>
- Dg2 is the face {2,3,4} = <<2>, <3>, <4>, <2,3>, <3,4>, <2,4>, <2,3,4>>

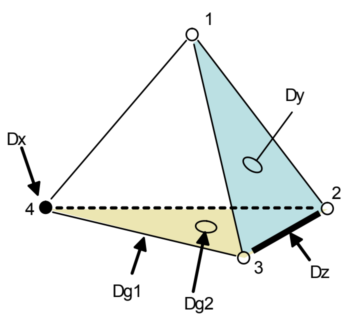

Figure 2. The Ground set Dg1 and Dg2.

■ Incidence matrix between two complexes

Given the complex K = K1 ⋃ K2, The incidence matrix I between the two complexes K1 and K2 is a 2D table having the description of K in terms of points, edges, faces etc. as rows and columns and where element

- I⟦i,j⟧ = 0 if the two complexes do not share the i and i subparts.
- I⟦i,j⟧ = 1 if the two complexes share the i and i subparts.

For example, with the two concepts defined above

- Dg1 is the edge {3,4} = <<3>, <4>, <3,4>>
- Dg2 is the face {2,3,4} = <<2>, <3>, <4>, <2,3>, <3,4>, <2,4>, <2,3,4>>

I	< 2 >	< 3 >	< 4 >	< 2, 3 >	< 3, 4 >	< 2, 4 >	< 2, 3, 4 >
< 2 >	□	□	□	□	□	□	□
< 3 >	□	1	□	□	□	□	□
< 4 >	□	□	1	□	□	□	□
< 2, 3 >	□	□	□	□	□	□	□
< 3, 4 >	□	□	□	□	1	□	□
< 2, 4 >	□	□	□	□	□	□	□
< 2, 3, 4 >	□	□	□	□	□	□	□

Note To be completed.

■ A distance between complexes

Redefinition of the black concepts Dx, Dy, Dz over the ground concepts Dg1 and Dg1 don't need any generic relations. We just redefine a black concept with *the "nearest"* Ground set concept.To do that, we have to compute some kind of distance Δ[x,y] between concepts, which amounts to a distance between complexes. This particular distance is computed by comparing the rows of bits for x and y in the $\mathfrak{K}$ table.

Distance definition For each couple of bits, $\{x_i, y_i\}$ where *xi* is the i-th bit of the row of concept x in the $\mathfrak{K}$ table (idem for y) a weight is computed using the following rules:

- if the two concepts have not the value ($x_i = 0$ $y_i = 0$) then the weight $w_i = 0$
- if one concept has the value and not the other ($x_i = 0$ $y_i = 1$ or $x_i = 1$ $y_i = 0$) then the weight $w_i = -1$
- if the two concepts have the value ($x_i = 1$ $y_i = 1$) then the weight $w_i = 1$

The weight function is summarized by the table $\mathfrak{W}$

$\mathfrak{W}$	0	1
0	0	-1
1	-1	1

Note other tables can be thought of. This one seems to work.

The distance is then defined by Δ[x,y] = $\sum_{i=1}^{n} w_i$ if there are n values (n columns in the $\mathfrak{K}$ table)

■ Example

Now we can create a distance table Δ with black concepts as rows and ground concepts as columns and in each cell x, y we put Δ[x,y].
Then we redefine concept Di as the ground concept Dg which has obtained the best Δ score:

Δ	Dg1	Dg2	
Dx	0	-1	⇒ Dg1
Dy	-2	0	⇒ Dg2
Dz	-1	1	⇒ Dg2

Note what if If all the scores of a Di domain row are < 0? we should not redefine the concept Di because there is too much difference with any of the ground concepts. (in the example, there is at least one ground concept with weight sum ≥0).

Note reversely, if there are more than one score >0 in a row we can redefine the black concept not as a single ground concept but as the list of ground concepts with scores >0.

■ Normalization of the distance

The best possible score (called BS) for an {x,y} couple of concepts is the max number of "1" of columns in the Ꮶ table. In the example, BS = 4.
This best score BS is used to normalize the absolute scores computed in the distance table Δ
We can say that
- Dx is a kind of Dg1 with -1/4 score (bad redefinition)
- Dy is a kind of Dg2 with 0 score (not good redefinition)
- Dz is a kind of Dg2 with 1/4 score (pretty good redefinition)

With that, pondered scores vary between -1 and 1.

Note we will abridge "is a kind of" in *isako* and write for example Dx $\xrightarrow{\text{isako}}$ Dg1.

■ Keeping track of the symbolic differences between concepts

When we say Dx $\xrightarrow{\text{isako}}$ Dg1, we loose the distance 'score' between Dx and Dg1. Moreover, we loose the information about the differences in terms of the subparts of the union complex.

For each redefinition, we can give the actual symbolic difference with
- a minus-list -{...} and
- an extra-list +{...}

of subparts (here, subparts are just values)
- Dx $\xrightarrow{\text{isako}}$ Dg1 - {3} +{}
- Dy $\xrightarrow{\text{isako}}$ Dg2 - {4} +{1}
- Dz $\xrightarrow{\text{isako}}$ Dg2 - {4} +{}

2.2 Working with tagged structures

■ Tagged structures

Tagged structures are described in chapter 1
We will consider successively three cases:
- simple fields
- one-level frames
- general frames

■ The case of fields

Fields are terminal tags, associated with a domain. For example, here are three fields T1,T2,T3 over the domains D1 and D2

```
T1 ≐ D1
T2 ≐ D2
T3 ≐ D2
```

Redefining a field consists in redefining the domain associated with the field. In other words, the "semantic thickness" of a field is not at the tag level but at the domain level.

Note several fields can be associated with the same domain (like T2 and T3 above); this is not a problem because it is the fields that are actually redefined.

We suppose that T1, T2, T3 are black concepts.
As far as domains D1 and D2 are concerned, we can't make any assumptions on them: they can be part of the Ground set or not. Three situations can arise

1. Di $\in \mathbb{G}$ and $\exists$ Tg $\in \mathbb{G}$ such that Tg $\doteq$ Di then Ti $\xrightarrow{\text{isako}}$ Tg

2. Di ∈ 𝔾 and ∄ Tg ∈ 𝔾 such that Tg ≐ Di then Ti $\xrightarrow{\text{isako}}$ Di

3. Di ∉ 𝔾 first the domain Di is redefined as a Dgi and (like in 2.) we have Ti $\xrightarrow{\text{isako}}$ Dgi

Note in situation 3. if there is no plausible Dgi in the Ground set for domain Di we have Ti $\xrightarrow{\text{isako}}$ {}. We can then define a special null domain DGnil ≐ EXT[{nil}] that is part of the Ground set, such that we'll have Ti $\xrightarrow{\text{isako}}$ Dgnil.

■ The case of one-level frames

One one-level frames are structured tags with all subtags that are fields.
We consider a simple case where the tagged structure to be redefined and the ground tagged structures are both one-level frames.

Let be three frames Tx, Ty, Ty defined over five different fields Ti = 1, 5 associated respectively with five different domains Di = 1, 5. This corresponds to the Tbox definition

```
{
 (* domains *)
 D1 ≐ def1, D2 ≐ def2, D3 ≐ def3, D4 ≐ def4, D5 ≐ def1
},
{
 (* one-level frames *)
 Tx ≐ {T1, T2},
 Ty ≐ {T1, T2, T3},
 Tz ≐ {T1, T2, T3, T4, T5},
 (* fields *)
 T1 ≐ D1, T2 ≐ D2, T3 ≐ D3, T4 ≐ D4, T5 ≐ D5
}
```

This Tbox can be illustrated with the following figure, with the tagged structure on the left and the corresponding simplicial representation on the right. As we are dealing with one-level frames, the vertices of the simplex can be directly associated with the domains symbols $T_{i=1,5} \longrightarrow D_{i=1,5}$.

We suppose that concepts Tx, Ty and Tz are black data to be redefined.
As Ti i=1,5 concepts are concerned, we can't make any assumptions on them: they can be part of the Ground set or not. We will suppose that T1, T3 and T5 are black data and that T2 and T4 are part of the Ground set. The same issue is raised by domains: we will suppose that domains D1, D2 and D5 are part of the ground set but not D3 and D4.

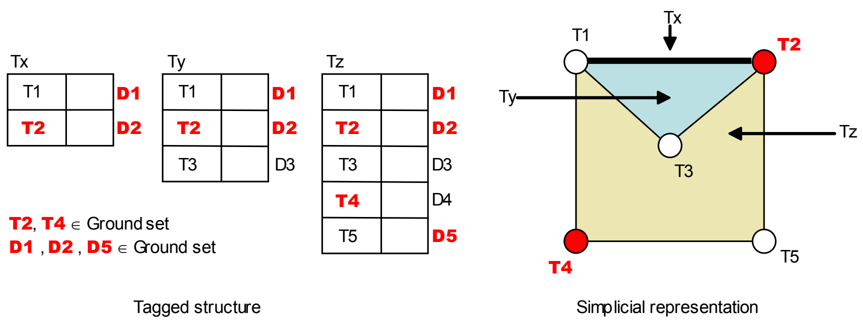

Figure 3. A tagged structure and its simplicial representation

Note In the figures, Ground concepts are shown with red-bold features. Small arrows denote the simplexes and bold arrows the isako redefinition operator.

When considering the fields, we are confronted with four situations (Ti is a field, 𝔾 the Ground set)

1. Ti ∈ 𝔾 Di ∈ 𝔾   no problem

2. Ti ∉ 𝔾 Di ∈ 𝔾   two subcases can occur:

a) ∃ Tg ∈ 𝔾 such that Tg ≐ Di then Ti $\xrightarrow{\text{isako}}$ Tg

b) ∄ Tg ∈ 𝔾 such that Tg ≐ Di then Ti $\xrightarrow{\text{isako}}$ Di

3.Ti ∈ 𝔾 Di ∉ 𝔾   Error: a tagged structure (field or frame) in the Ground set must be fully understandable so its associated domain(s) must be in the Ground set too.

4. Ti ∉ 𝔾 Di ∉ 𝔾   First we redefine the domain Di (see section on redefining domains) then we go to situation 2.b)

Once we have dealt with the fields, we can see that field symbols play exactly the same role as value symbols in the previous section. So the redefinition of a one-level tagged structure is quite the same as an extensional domain. Below, on the left, is shown the field redefinition process and on the right the subsequent process for the one-level frames (se suppose that the distance scores are not negative, thus every concept has a plausible Ground set counterpart)

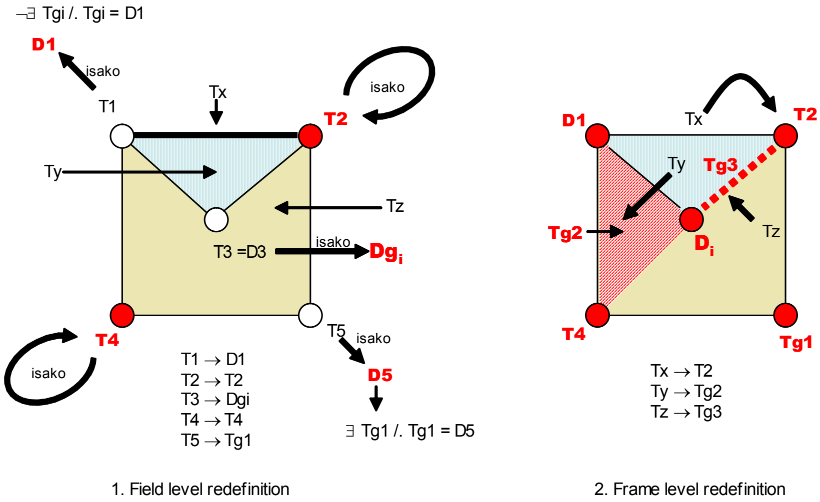

Figure 4. Redefinitions operations at the field and frame level.

▪ The case of general frames

We will consider a two-level tagged structure

```
Tbox1 = Sequence[{
   (* domains are simplified *)
   D1 ≐ EXT[{1}], D2 ≐ EXT[{2}], D3 ≐ EXT[{3}], D4 ≐ EXT[{4}], D5 ≐ EXT[{5}]
  },
  {
   (* two-level frames *)
   Tx ≐ {Ta, Tb},
   Ty ≐ {Ta, Tb, Tc},
   (* one-level frames *)
   Ta ≐ {T1, T2},
   Tb ≐ {T2, T3},
   Tc ≐ {T1, T2, T3, T4, T5},
   (* fields *)
   T1 ≐ D1, T2 ≐ D2, T3 ≐ D3, T4 ≐ D4, T5 ≐ D5
  }];
```

This Tbox can be displayed as a hierarchy of boxes like in the figure below

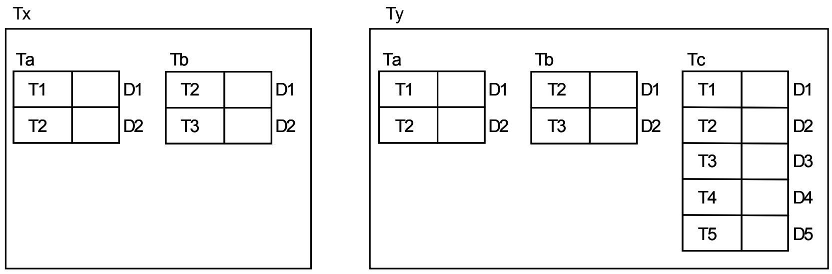

Figure 5. A two-level tagged structure.

The corresponding simplicial representation is then

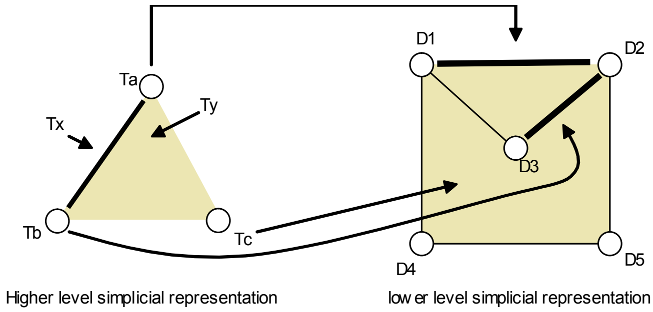

Figure 6. Simplicial representation of the two-level tagged structure

Note at the field level, is indifferent to represent a field by its tag symbol or by its domain symbol. In the figure above right we chose to label the simplicial nodes with domains' names D1, D2, ..D5.

For each level of the tagged structure a simplicial representation is associated, so that vertices of level n are given a simplicial representation at level n-1; this is repeated until the field level is reached.

Principle of multilevel redefining the tag concepts are redefined, beginning with the lowest level

1. the domains are first redefined: $Di \xrightarrow{\text{isako}} \mathrm{Dg}i$
2. the fields Tf are then redefined with the redefined domains $\mathrm{Tf} \doteq Di \xrightarrow{\text{isako}} \mathrm{Tf} \doteq \mathrm{Dg}i$
3. the one-level tags T1 are redefined with the redefined fields $\mathrm{T1} \doteq \{\mathrm{Tf}_1, \mathrm{Tf}_2\} \xrightarrow{\text{isako}} \mathrm{Tf} \doteq \{\mathrm{Tf}_1 \doteq \mathrm{Dg}i, \mathrm{Tf}_2 \doteq \mathrm{Dg}_j\}$

...

n. until the highest level tags are redefined.

The step 3. of the multilevel redefining principle, applied upon the current example, is illustrated in the figure below. We will suppose that the domain level has already been processed (we could say for simplification, that domains Di are part of the Ground set).

- the one-level tags are redefined as stated in the previous section (figure left)

$\mathrm{Ta} \xrightarrow{\text{isako}} \mathrm{Tg1}$

$\mathrm{Tb} \xrightarrow{\text{isako}} \mathrm{Tg2}$

$\mathrm{Tc} \xrightarrow{\text{isako}} \mathrm{Tg3}$

- then the definition of Tx and ty are expressed with Tgi (figure right)

$\mathrm{Tx} \xrightarrow{\text{isako}} \{\mathrm{Tg1},\mathrm{Tg2}\}$

$\mathrm{Ty} \xrightarrow{\text{isako}} \{\mathrm{Tg1},\mathrm{Tg2},\mathrm{Tg3}\}$

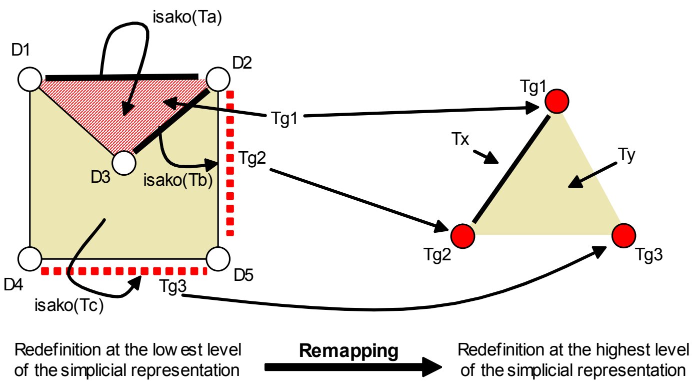

Figure 7. Multilevel redefining.

In this example, each subtag has been given an isako redefinition, meaning that we have found a plausible ground concept for each lower level concept. Now, suppose that tag Tc has a negative score so that it makes it impossible to find a plausible redefinition tag within the ground concepts: Tc $\overset{\text{isako}}{\longrightarrow}$ {}. In this case, there is a loss of dimension for the higher concepts supported by Tc, like Ty concept that is restricted to Ty $\overset{\text{isako}}{\longrightarrow}$ {Tg1,Tg2}. Then we have the same redefinition for Tx and Ty making them synonyms within the Ground set. The next figure illustrate this case.

In order to remember that there has been a loss in dimension/information, we can add to every ground set a null concept (called Tgnil) that allows to re-extend the Ty concept like this: Ty* $\overset{\text{isako}}{\longrightarrow}$ {Tg1,Tg2,Tgnil}, so that it can't be confused with Tx. Below, the re-extension is denoted with dotted lines.

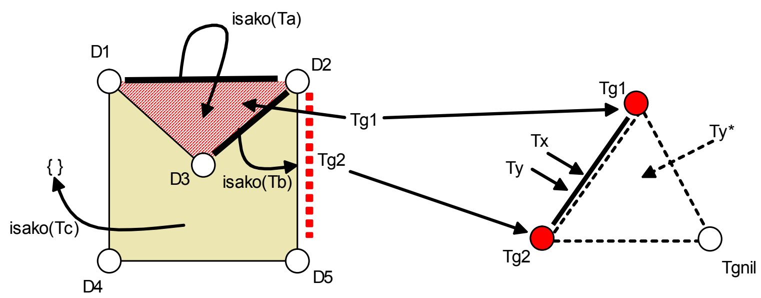

Figure 8. Introducing the Tgnil concept.

Note Tgnil can be defined as Tgnil ≐ Dnil with Dnil≐ EXT[{nil}] where the "nil" value is also part of the Ground set. In assertions, the fields associated with Tgnil have always nil as a value.

2.3 Working with assertions

Let be the following Abox1, consistent with Tbox1 defined above, where all assertions are named x_i which are considered as black data.

```
Abox1 = {
  x1 ≐ T1[1],
  x2 ≐ T2[], (* underspecified *)
  x3 ≐ Ta[T1[1], T2[2]],
  x4 ≐ Tx[Ta[T1[], T2[]], Tb[T2[2], T3[3]]](* containing underspecifications *)
 };
```

If the an agent has to explain x1, x2, ... it must redefine the expressions associated with the x_i. As these expressions are quite similar to tagged expressions, especially when there are underspecifications, this comes down to the redefinition of tagged structures. This demonstrates that semantic heterogeneity is to be handled at the Tbox level, not at the Abox level.

3. Polygonal paths

3.1 Definition

See Erika's Thesis for a definition of polygonal paths in the simplicial theory.

3.2 Polygonal paths for domains

▪ Source and target concepts

Definitions

Source concept in the Black data set we consider a *distinguished* concept that we want to explain, i.e. to redefine. This concept is called the *source concept* or the *black source* (Bs).

Target concept in the Ground set we consider a *distinguished* concept that we want to use as a support for redefinition (for any particular reason). This concept is called the *target concept* or the *Ground target* (Gt).

Let be the domains

```
D1 ≐ EXT[{1, 2, 3}];(* source *)
D2 ≐ EXT[{2, 3, 4}];
D3 ≐ EXT[{1, 4}];
D4 ≐ EXT[{3, 4, 5, 6}];
D5 ≐ EXT[{5, 6, 7}];(* target *)
```

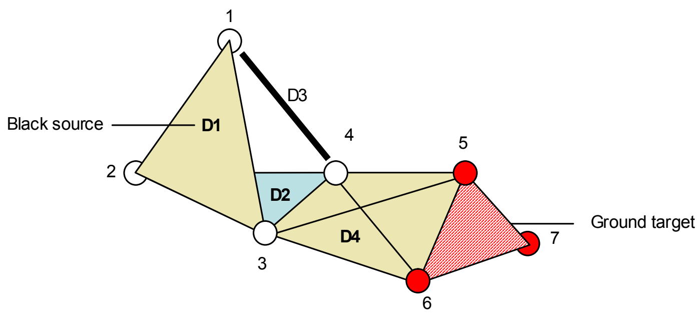

Figure 9. Source and Target concepts are not directly connected.

We'll suppose that D1, D2, D3 and D4 are black concepts (among them we are particularly interested in redefining D1) and that D5 is the item of the Ground set in which we are particularly interested. With the definitions above:

- D1 is the source concept
- D5 is the target concept

When applying the previous redefinition algorithms for domains, we obtain

D1 $\xrightarrow{\text{iskao}}$ {} with score -2

D2 $\xrightarrow{\text{iskao}}$ {} with score -2

D3 $\xrightarrow{\text{isako}}$ {} with score -1.66

D4 $\xrightarrow{\text{iskao}}$ D5 with score 0

This shows that D1 cannot be redefined in terms of D5. The algorithm fails. Moreover, it fails also for D2 and D3 because in the given situation, the connectivity of the simplicial representation of the domains is scarce. This situation occurs quite often. Therefore we have to prepare to offer degraded solutions to the redefinition problem. To achieve that, the *polygonal paths* will be used as a support.

Now, if we consider that D4 is some sort of a degraded Ground concept (i.e. an approximation of D5) we will have to redefine the remaining domains in terms of D4

$D1 \xrightarrow{\text{isako}} D4$ with score -1.5

$D2 \xrightarrow{\text{isako}} D4$ with score 0.25

$D3 \xrightarrow{\text{isako}} D4$ with score 0.

This time, two concepts have a non negative score redefinition: D2 and D4. Note that admittedly D1 is attained but with a very bad score. We cant redefine directly D1 in D4; it is more reasonable "to pass via" D2 or D3. This leads to two possible ways:

1. if we consider that D4 is some sort of a degraded Ground concept (an approximation of D4 which is in turn is an approximation of D5) we will have

   $D1 \xrightarrow{\text{isako}} D2$ with score 0

2. if we consider that D3 is some sort of a degraded Ground concept (an approximation of D4 which is in turn an approximation of D5) we will have

   $D1 \xrightarrow{\text{isako}} D3$ with score - 0.5

The sums of the scores along the 3 possible polygonal paths between D1 and D5 are:

<D1, D4, D5> = -1.5 + 0.25 + 0 = -1.25

<D1, D2, D4, D5> = -0.33 + 0.25 + 0 = -0.08

<D1, D3, D4, D5> = -0.5 + 0 + 0 = -0.5

Consequently, it is reasonable to choose the path <D1, D2, D4, D5> with path score quite near 0 (-0.08).

■ Algorithms

This lead to two opposite strategies for redefining a source concept Bs along a polygonal path towards a target concept of the Ground set Gt. We can choose to redefine a concept Bs along

1. the *shortest* path to Gt thus minimizing the loss of information by successive redefinitions. In this case, we only consider transitive connected without regarding the connectivity score at each step (or when two paths are of equal length using the connectivity score as a second discriminating criterion).

2. the *thickest* path to Gt thus minimizing the loss of information by redimensionalization. In this case, we only consider the best possible connectivity at each step without regarding the path length (or when two neighbors are of equal connectivity using the path length as a second discriminating criterion).

Note Thickness can be defined more precisely as the best adequacy between two concepts, that is the ratio of the subparts they share proportionate to their dimension. In other words, the thickest neighbor is not the neighbor with the highest dimension but the neighbor with the maximum relative sharing.

Algorithm: *shortest_path(Bs_source, Gt_target)*

start by putting all nodes but Gt in the *sourcebag* and Gt in the *targetbag*

set Bs as the *source* and <Gt> as the *path*

while the sourcebag is not empty **do**

  foreach couple of nodes (s,t) ∈ sourcebag × targetbag

  Apply domain_score(x, t)

  if ∃ t ; domain_score(source, t)≥ 0 [1] **then** append t to the path and **exitwhile**

  else remove from the sourcebag all nodes with a score ≥ 0 and put them in the targetbag

  endwhile

if the sourcebag is empty **then return** *failed* **else return** the path

[1] when there are several targetbag concepts t such that score[x,t) ≥ 0, a second criterion can be applied in order to discriminate them, like the thickness defined in the *thick_path* algorithm. Another policy would be to construct as many subpaths as there are equal local scores and sort them by length at the end.

Note domain-score[x_, y _) is the score function over domains defined above.

Generalization this algorithm is easy to generalize from a single target concept to a full Ground set by putting not only Gt but all the Ground set in the targetbag and by developing as many paths as concepts in the Ground set.

Algorithm: *thickest_path(Bs_source, Gt_target)*
start by putting all nodes but Gt in the *bag*
set Gt as the *target* Bs as the *source* and <Bs> as the *path*
while the bag is not empty **do**
foreach couple of nodes (source, x) ; x ∈ the bag
apply domain_score(source, x) and let be t_{best} the t node with the best score [1]
remove t_{best} from the bag and append it to the path
If t_{best} = Gt **then exitwhile else** set t_{best} as the source
endwhile
if the bag is empty **then return** *failed* **else return** the path

[1] when there are several best score concepts t, a second criterion can be applied in order to discriminate them, like the distance to Gt as defined in the *shortest_path* algorithm. Another policy would be to construct as many subpaths as there are equal local scores and sort them at the end using the summation scores over the paths.

Note domain-score[x_, y _) is the score function over domains defined above.

Generalization this algorithm is easy to generalize from a single target concept to a full Ground set by putting not only Gt but all the Ground set in a targetbag and replacing the test t_{best} = Gt by MemberQ[targetbag, t_{best}].

The algorithms can fail. This occurs when either the source concept or the target concept are *disconnected* from the other concepts of the Kbase (or belong to a disconnected subset of concepts). In the simplicial representation, this situation corresponds to a complex with disconnected subparts like in the following figure

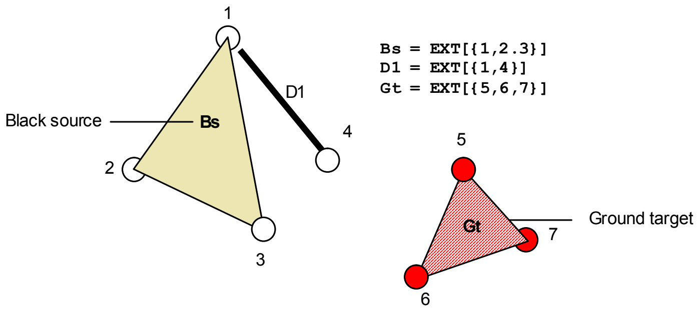

Figure 10. Source and Target concepts are disconnected.

3.3 Polygonal paths for tagged structure

■ Basic principles

The polygonal paths approach can be generalized from domain to tagged structures.
In this case, we have a black tagged structure to redefine and a Ground set containing target concepts of any levels. The tagged structure are of two kinds:

Field structures black fields can be explained by simply redefining their associated domain using the *shortest_path* or the *shortest_path.*algorithms.

Frame structures black frames contain subtags that can be part or not of the Ground set.
- by principle of the simplicial representation, if any tag is part of the Ground set all its components (subtags or fields) must be part of the Ground set. Consequently, subtags that are part of the Ground set don't need to be redefined.
- black subtags need to be redefined. In turn, they can contain subsubtags that are part or not of the Ground set.

In order to redefine black frames, or subframes, two main policies can be envisioned:

Breadth-first apply the polygonal paths algorithms proposed above with tags symbols acting as domain symbols and if the path contains black subtags redefine them recursively.

Depth-first go down to the level of fields (terminal tags) and apply the polygonal paths algorithms above the associated domain.

▪ Breadth-first paths

Example

```
Ts ≐ {T1, T2, T3}(* source concept *)
Ta ≐ {T2, T3, T4}
Tb ≐ {T3, T4, T5}
Tg ≐ {T4, T5, T6}(* target concept *)
T1 ≐ ...,
..., T6 ≐ ...
```

This situation is illustrated in the figure below

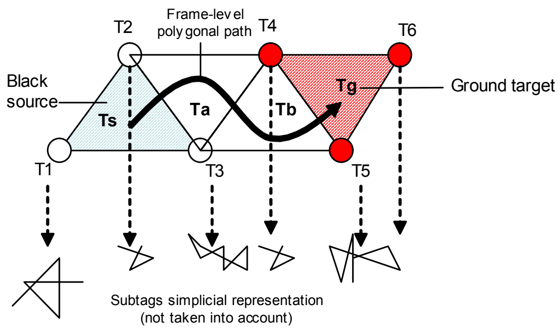

Figure 11. Breadth-first path from Source to Target.

In this example, without regarding the subsubtags, the redefinition is associated with the top level polygonal path which would produce for Ts the following path

$$\mathbf{Ts} \xrightarrow{\text{isako}} < \mathbf{Ta}, \mathbf{Tb}, \mathbf{Tg} >$$

The problem is that the high level path can contain black tags along the path. For example, Ta or Tb or subtags of Ta or Tb can be black concepts (indeed, they must be or else Tg would not be the target!). Then, they must be redefined in turn with the same policy: they are redefined and if their path contains black data, the algorithm is applied recursively until fields are encountered; black fields can be explained by simply redefining their associated domain using the *shortest_path* or the *shortest_path* algorithms.

Algorithm: *breadth_first_path(Ts,Tg)*
let be Ts the *source* and Tg be the *target*
if Ts is a field
return the *shortest_path(Ds,Dgset)* or the *shortest_path(Ds,Dgset)* where Ds is such that Ts≐Ds and Dgset is the set of all Ground set domains (generalized version of the algorithm — see note).
else
convert the definition expression tag lists of all tags into extensional-definition like domains:
∀ Ti ; Ti ≐ {T1, T2, ...} ⇒ Di ≐ EXT[{x1, x2, ...}] ; xi being viewed as atomic values
Apply the *shortest_path(Ds,Dg)* or the *shortest_path(Ds,Dg)*
reconvert the resulting path <Da, Db, ... Dg> into <Ta, Tb, ... Tg>
Foreach black tag t of the path **do**
apply recursively *breadth_first_path(t,Tg)*
return the path with subpaths associated with black subtags <Ts, <Ta, ...Tg>,... Tg>
endelse

■ **Depth-first paths**

to do

4. Adaptation

4.1 Redefinition equations

■ **Semantic heterogeneous equations**

When two agents A1 and A2 share a common Ground set and that agent A "explains" one of its black concepts Bi to agent A2 so that A2 can "understand" it, we have defined precisely what "explain" is

explain(Bi) = redefine(Bi) in terms of the Ground set, i.e. Bi $\overset{\text{isako}}{\longrightarrow}$ Gi, choosing between several policies.

but we have not defined precisely what "understand" is. Because Gi is part of the Ground set, it is fully part of the Kbase of A2, so it is fully understood by A2. Hence, one cannot say that Bi = Gi but Bi = *iskao(Gi, + indications of differences)* then we state

understand(*iskao(Gi, + indications of differences)*) = find within the black data of A2 a black concept Bj such that Bj has the same differences with Gi than Bi had with Gi.

This can be illustrated by the following scheme

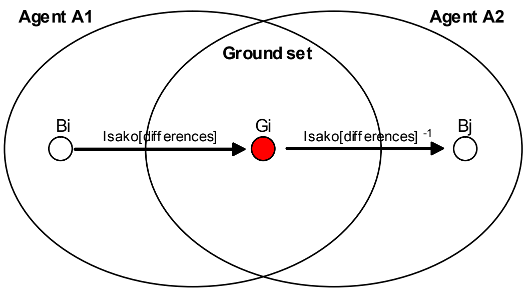

Figure 12. Projecting a redefinition path on the other side.

This situation is analogous, but not quite the same, as in Case Based Reasoning (CBR) where we have a scheme "D is to C what B is to A". In our case, Gi plays both the role of C and A.

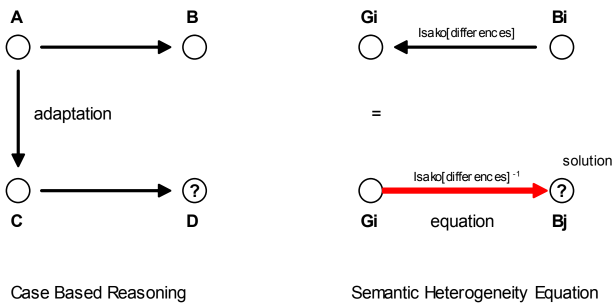

Figure 13. Case-based Reasoning principle compared to SH equations.

■ **Example with domains**

Let be the following agents' world with two agents A1 and A2 sharing a Ground concept Dg

```
(** Agent A1 **)
Dg ≐ {2, 3}
 Ds ≐ {1, 2, 3}

 (** Agent A2 **)
 Dg ≐ {2, 3}
 Dx ≐ {2, 3, 4}
 Dy ≐ {3, 4}
 Dz ≐ {4, 5}
```

This is illustrated by

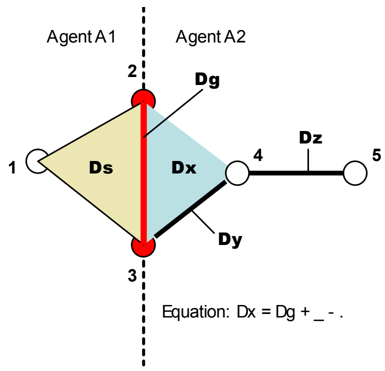

Figure 14. A simple SH equation.

If we redefine Ds of agent A1 we obtain

```
Ds == isako[Dg, add[_ "1"], del[]];
```

meaning that Ds is a Dg to which we should *add* an unknown node (denoted _) and delete (*del*) no nodes (denoted []). This equation defines a pattern over the concepts of agent A2 of the form

```
PATTERN == DOMAINEXTENSION[Dg] ⋃ {_ "1"} == {2, 3, _ "1"}
```

When this PATTERN is applied to the DOMAINEXTENSION of the conceptlist of A2 the best solution is concept Dx.

```
ADAPT[{Dg ≐ {2, 3}, Dx ≐ {2, 3, 4}, Dy ≐ {3, 4}, Dz ≐ {4, 5}}, {2, 3, _ "1"} ]
⟹ {2, 3, 4} ≐ Dx
```

while making the assumption that "1" should be viewed as a 4 value. Therefore, each time an adaptation is made, it introduces automatically subsequent correspondences between heterogeneous concepts. Such assumptions, are added to the common Ground set between agent A1 and agent A2 and will be used in further heterogeneous communication, so they can stand if successful or else be challenged if they provoke a lot of inconsistencies; in the later case, there can be a backtrack and the previous adaptation erased and or modified.

4.2 Direct adaptation

■ An abridged notation

Let be two agents with their Kbases represented by simplicial complexes.
- The nodes are considered atomic concepts (e.g. values or closed frames) and are denoted
  - ai for agent A
  - bi for agent B
  - gi for the ground {A,B}
- The structures (e.g. extensional domains or frames) are denoted
  - Ai for agent A
  - Bi for agent B
  - Gi for the ground {A,B}

Here is an example of a classical WAP situation between A and B using this notation

```
G ≐ {g1, g2}
A1 ≐ {a1}
A2 ≐ {a1, a2}
A3 ≐ {a2, g1, g2}
B1 ≐ {b1, g1, g2}
B2 ≐ {b1, g2}
B3 ≐ {b1, b2}
```

The two Kbases have been unioned but is very easy to visualize the Ground set and the black data of each agent (each set has symbols beginning with its associated agent-letter but the ground which is in "Gs")

```
Groundset[A, B] = {g1, g2, G}
Blackdata[A] = {a1, a2, A1, A2, A3}
Blackdata[B] = {b1, b2, B1, B2, B3}
```

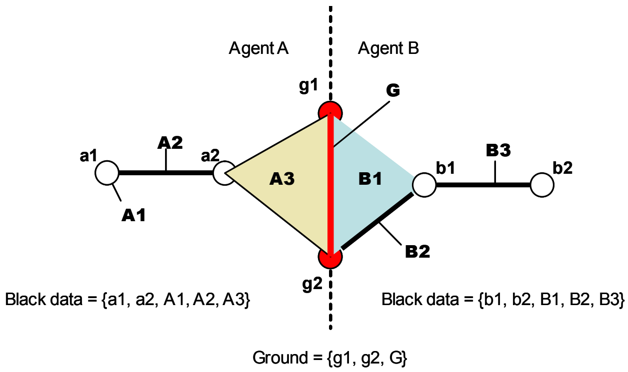

Figure 15. Simplicial representation of the example.

■ Redefining nodes

The black data is constituted of nodes ai/bi and structures Ai/Bi. We know that considering the dual simplicial representation nodes and structures can be seen as dual too. This makes it possible to explain basic entities such as ai/bi in terms of the structures Ai/Bi they appear into. Here is the dual representation

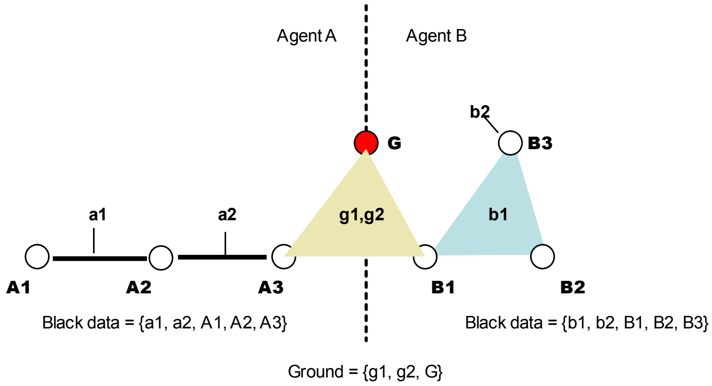

Figure 16. Dual representation of the figure 15.

Note the two basic entities of the Ground set (g1,g2) are synonym in the dual representation. It comes down to having only one entity for the support of redefinition which is not a loss in information but in the power of handling heterogeneity.

Note Black data and the Ground set remain the same.

The ex-nodes a1, a2 or b1, b2 are now structures. They can be handled exactly with the same algorithms proposed for domains and frames, the ex-structures Ai, Bi playing the part of nodes.

5. Complements

5.1 Comparison between graph paths and polygonal paths

We have seen two main approaches to concept redefinition
- the lexical SH based on graph path redefinition,
- the structural SH based on polygonal path redefinition.

Graph paths, in turn can be approached in two ways
- a simple un-labelled graph of concepts, by not taking into account the relation semantics, or in other words, where all relations are considered the same,
- a labelled graph of concepts, where labels are generic relations, known by all agents.

In the first case of graph paths, a redefinition path between a black source concept Bs of a Tbox1 of an agent A1 and a Ground set concept Gi is of the form: < Bs → B1→ B2→ Gi > or by analogy with file paths Bs/B1/B2/Gi where B1, B2, etc. are black concepts of Tbox1 that are of no further use. When we will try to build the image of such a path in the Tbox2 of another agent A2 sharing Gi, we will obtain Gi /{..} / {..} / {X..}. which means that every concept at a distance of k = path-length[Bs/B1/B2/Gi] is image of Bs in Tbox2. This approach to redefinition is too vague to be of any interest. This illustrated in the next figure

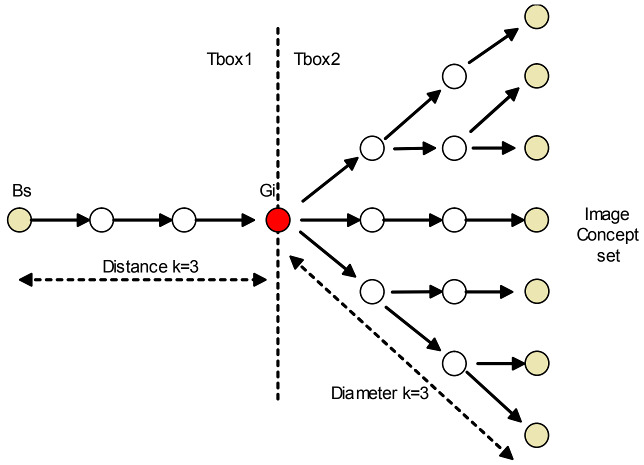

Figure 17. Projection of a simple graph path.

In the second case of graph paths, a redefinition path between a black source concept Bs of a Tbox1 of an agent A1 and a Ground set concept Gi is of the form: $< B \xrightarrow{a} B1 \xrightarrow{b} B2 \xrightarrow{c} Gi >$ where a, b, c etc. are some generic relations and B1, B2, etc. are black concepts of Tbox1 that are of no further use. When we will try to build the image of such a path in the Tbox2 of another agent A2 sharing Gi, we will obtain $< Gi \xrightarrow{c^{-1}} \{..\} \xrightarrow{b^{-1}} \{..\} \xrightarrow{a^{-1}} \{X..\} >$. In this case, the relations symbol add constraints at each step of the path and not every concept at distance k is image of Bs; besides, it can happen that there is no image at al Bs in Tbox2; We are in the opposite situation where the redefinition is often too straight. This illustrated in the next figure

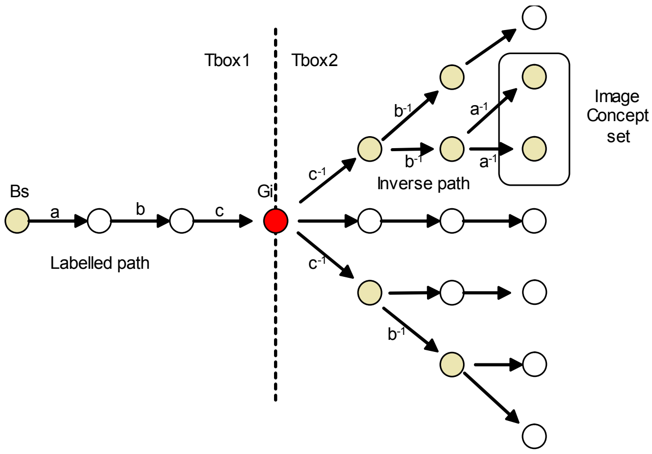

Figure 18. Projection of a labelled graph path.

Now, if we consider polygonal paths, relations are not any more represented by symbols but instead by their extensions. This provides two advantages
- the semantics of relations is implicitly represented by the extensions, so it is taken into account, implicitly at least, whereas in the lexical approach we rely only on the difference between the relations symbols (viewed then as mere labels).
- as extensions can be of some size the polygons can be of high dimension. In this case, when we compare the distance between complexes we can have a relatively smooth function that is favorable to approximations.
This illustrated in the next figure

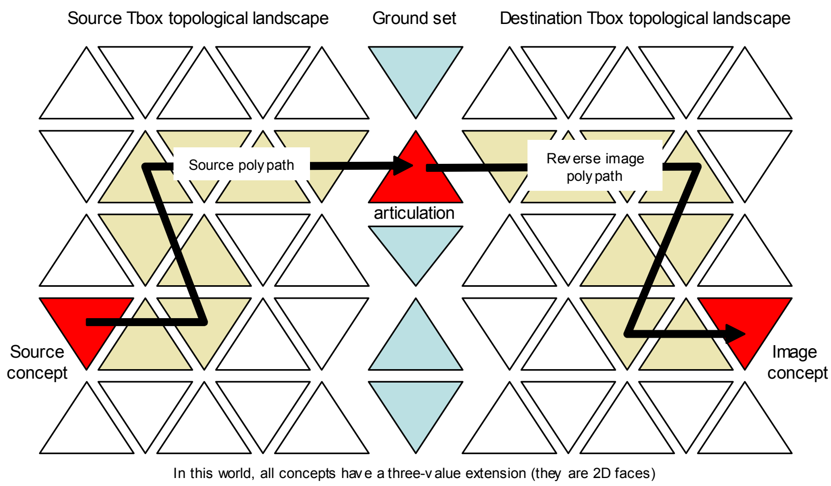

Figure 19. Projection of a polygonal path.

Once a polygonal path between a source concept and a ground concept has been defined, we dispose of a topological structure of which we want to compute the reverse image within the destination Tbox staring from the ground set articulation and leading to the image concept at the end. When an exact image does not exists, we want to compute the best approximate image of the source path.

5.2 Cross categorization

■ Social Categorization

We have supposed so far that agents use categories, mostly as domains, or tagged structures. But we did not rise the issue of how the agents created those categories in the first place. Somehow, entities seem to pop pup into the agents, and we are not supposed to make enquiries about their genesis.

Concerning the genesis of concepts, three main situations are possible

internal genesis the agent is considered isolated from other agents and any environment. In this situation, it is difficult to envision any genesis; we are doomed to resort to axiomatic definition.

autonomous genesis the agent is considered isolated from other agents but interacts with an environment in which it is situated. In this situation, concepts can be learned from the interactions with the environment (problematical of situated perception).

social genesis the environment of an agent is constituted of other agents. In other words, there is a distinguished agent within a population of so-called informational agents. In this situation, concepts can be learned from the interactions with the other agents.

If we place ourselves in the social genesis situation, of interest when informational agents are considered, two main situations can arise

concept exchange supposing agent A1 knows a concept C^1 that agent A2 don't know, then agent A1 "explains" C^1 to agent A2, using redefinition techniques as suited in the previous chapters.

concept cross categorization supposing agent A1 knows a set of concepts C_i^1 and agent A2 knows a set of concepts C_i^2, such that $C_i^1 \cap C_i^2 = \emptyset$, but the extensions of those concepts have a non empty intersection, then that intersection can serve as a basis for categorization of new concepts for agent A1 and B1. Such a categorization is then relative to the couple {A1,B1}. In a population of agents, such a categorization can be relative to groups of agents {A1,A2,A3, ...} when one consider intersections over the group.

■ **Principle of cross categorization**

Supposing there are two agents A1 and A2, such that
- agent A1 knows a set of concepts C_i^1
- agent A2 knows a set of concepts C_i^2, such that lexically $C_i^1 \cap C_i^2 = \emptyset$
- the extensions of those concepts have a Ground set in the form of a non empty intersection.

Then agents can exchange their concepts through the Ground set. Doing that, whereas the Ground set was of no conceptual interest *before* the formation of the interactional couple {A1,A2} — it just didn't exists conceptually — *now* the elements of the Ground set are put in focus up to their possible reification/categorization as mutually categorized concepts between A1 and A2. *After* that, the reified concepts are used as a common language for further interactions between A1 and A2. Moreover, if the interaction of A1 and A2 becomes permanent it can be very useful for agents A1 and A2 to redefine all their concepts in terms of the Ground set reified concepts which act then as a mutual conceptual basis for categorization of everything.

Before the formation of the {A,B} couple we are in the following situation

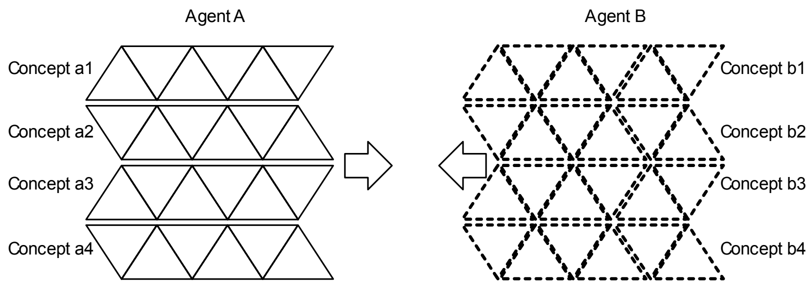

Figure 20. Before the encounter of two aliens.

After the couple {A,B} is formed, a Ground set is exhibited and its subparts tend to be reified, that is associated with new symbols that are shared by the two agents

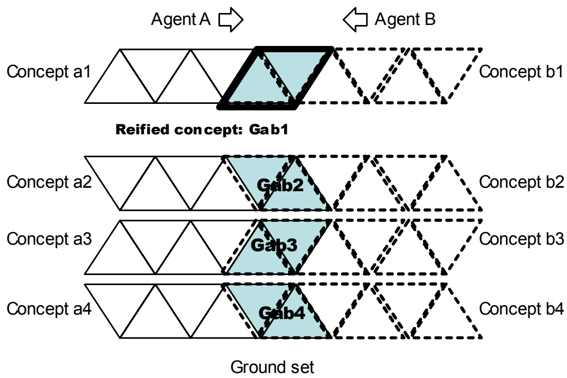

Figure 21. The encounter of two aliens generates a Ground set.

In the next stage, the new Ground set concepts are used as a base for expressing all the concepts of both agents A and B

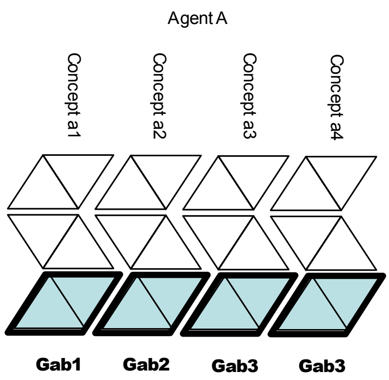

Figure 22. Each alien redefines all its black data in terms of the Ground set.

■ Table of Contents

5 Complements

5.1 Comparison between graph paths and polygonal paths

5.2 Cross categorization

6 Table of Contents