Visualization needs search context:

Ontology = In N Out Menu

Root = In N Out Menu

“In the context of IN N OUT MENU”

=

Root = menu = hole in the middle = 0, 0, 0

Layer 1 = menu items = 4 = regions of vertices = x, x, x

Layer 2 = menu item recipe = grid in regions = x.y, x.y, x.y

Biggest layer, how to calculate? Search parameters -> Layer 3 = recipe variations = smaller divisions grid in regions = x.yz, x.yz, x.yz

Layer 4 = recipe ingredients, themselves = smaller parts of the grid in the regions = x.yzz1, x.yzz1, x.yzz1

Next ontology = American Fast Food Ingredients

CB is just like neo4j visualized in 3D with the ability to realize new objects that instantiate the ontology but were not previously in it, through applying X to the search context via matrix multiplication, which sends the core identity traits of X (salad) to the boundaries of the search context (in n out menu), such that it outputs a specified search result “if there was an In N Out Menu object in the mode of salad that didn’t change any ingredients of the In N Out Ontology it would have the traits and superposition of acceptable spectral value ranges of…”

Return:  
 “How many ingredients in the In N Out Ontology also satisfy the spectral values of the parameters of the core identity traits of salad?”

{Bun, vegetables (chilies, lettuce, tomato, onion, pickle), mustard, mayonnaise, ketchup, beef, cheese}

Then  
Check how many of these satisfy the traits of salad by themselves and are self-sufficient and are salads

Salads = {vegetables} = (5)

implies

= bun, mustard, mayonnaise, ketchup, beef, and cheese do not make salads by themselves and are non-self-sufficient and non-salads

Implies

For all vegetables, each has variations with only 1 of each 6 other non-vegetable ingredients (excluding the other “satisfies the traits by itself” pairs)

Then   
Search parameters open again and user specifies the spectral boundaries for the self-sufficient and non-self-sufficient ingredients

Then  
We can break the matrix multiplication into two:

1. Let there be a function to calculate the matrix of self-sufficient salads with non-self-sufficient non-salads in the ontology according to the spectral boundaries   
     
   f(x) = self-sufficient objects in n out menu salads(non-self-sufficient in n out menu non-salads)
2. Let there be a function to calculate the matrix of self-sufficient salads with self-sufficient ingredients and non-self-sufficient non-salads according to the spectral boundaries  
     
   f(x) = self-sufficient in n out menu salads(self-sufficient in n out menu salads, non-self-sufficient in n out menu non-salads)

Then   
Search parameters open again and user specifies  
 Then

Repeat

Then

1. Let there be a graph space where “In the context of IN N OUT MENU” =

Unrealized Hypergrid of “identifiable objects”

* 1. Root: f(x) = in n out menu(salad) = hole in the middle = (0, 0, 0, 0, 0)
  2. Layer 1 = menu items, self-sufficient = 5 = regions of vertices = objects 1-5 regional coordinates
     1. Region 1 = mega-grid coordinates (1.0, 1.0, 1.0, 1.0, 1.0)
  3. Layer 2 = menu item recipe, non-self-sufficient due to being parts = Calculations 1 and 2 = grid in regions
     1. deca-grid coordinates (1.1, 1.1, 1.1, 1.1, 1.1)
  4. Layer 3 = recipe variations = smaller divisions grid in regions = x.yz, x.yz, x.yz
     1. centi-grid coordinates (1.11, 1.11, 1.11, 1.11, 1.11)
  5. Layer 4 = recipe ingredients, themselves = smaller parts of the grid in the regions = x.yzz1, x.yzz1, x.yzz1
     1. milli-grid coordinates (1.111, 1.111, 1.111, 1.111, 1.111)

Unrealized Hypergrid of “identifiable objects in the context of current outermost layer”

Next ontology = American Fast Food Ingredients

* + 1. = root = recipe ingredient itself (previously z1)
    2. Layer 1 = all cuisines its found in

OR

PREMISE:

You want to search

So you get to see the general object (the root) of what you are looking for

And then you specify it yourself.

///////////////////////////////////////////////////////////////////////////

MECHANICS:

If we number all of the nodes and relationships in the ontology before we even try to realize them then we are able to calculate the matrix \*without calculating the exact properties individual objects\* and we can know not only how many objects will result from the calculation and therefore always minimize the processing,

But also we can AUTOMATICALLY REALIZE the space of intellectual objects (superposition) as strings of numbers and those numbers can be ASSIGNED (NOT CONVERTED) a COORDINATE SET and displayed in a 2/3D graph space, REGARDLESS OF IF THAT INTELLECTUAL OBJECT (SUPERPOSITION) EXISTS AS A POSE IN THE CURRENT ONTOLOGY

And each one of those coordinate sets will contain one number that is strange (like 12.112312351, 1, 3) and that coordinate and all the others (12.1x, 1, 3) represent similar things.

Then, by knowing which numbers correlate to which types of things, we can calculate, search for, and display (realized) the actual properties of ANY POSE from the superposition, by converting the string back from numbers to ontological objects without doing any of the other calculation, and display that.

///////////////////////////////////////////////////////////////////////

First you synthesize the salad object with CITs

Then you synthesize the In N Out Menu with CITs

“In N Out Menu”

“+menu item ‘salad’”

“Salad CIT = …”

APPLY -> this realizes the superposition in the blackbox graph and ports it to the display as a interactive regionalized torus (or tori)

INVESTIGATE -> Specify Display of In N Out Salad Options

“In N Out Salads with a lettuce base”

“In N Out Lettuce-based salads with maximum 30% patty and no cheese”  
 “With no croutons or onions”

“With tomato”

“And secret sauce”

NOTES:

In N Out ontology arrows should have “all possible orders” between menu recipes (hamburger, plain hamburger, cheeseburger, cheeseburger with onion and mustard, etc) and ingredients themselves