the entire point is that Crystal Ball has dynamiccontext and is updating its skills based on rules from the context values.

then, it is building an ontology from that dynamic context, to get a deeper understanding of intent

then, it is using the attributes of the deep context to assemble a proposed answer, which is encapsulated in the informatihedron's attributes (properties and boundaries)

properties are any quality

boundaries are limits to the qualities of the qualities

then, it is using the informatihedron as an output, and the user uses it as an input to refine by responding with more desired properties and boundaries

then, it is mining those properties and boundaries by assembling dynamic skillchains to traverse the domain space effectively and search for combinations between other nodes and the informatihedron. it always keeps the informatihedron as the current node, and only moves to nodes that add attributes, either properties or boundaries requested by the user.

then, when the user accepts it, they instance it

when it's instanced, it becomes an object in a neighborhood representing the class of instances of correct answers, so as an instance informatihedron of the answer instance class, the user can use a different type of interaction:

the output is: this is the informatihedron, and these are the 3 nearest neighbors

the user can then request filtration of unique patterns only (potential novel emergence)

OR

add more boundaries to constrain them