

Ontologies and Conceptual Metaphor in Autonomous Robotics

David Dodds Open-Meta Computing Inc david_dodds_2001@yahoo.com

Abstract—The computer programs which control autonomous mobile robots can make use of fuzzy processing and computing with words in order to extend their capabilities by allowing them to be less literal and exact match oriented. Both sensor data interpretation and planned activities subtleties benefit from fuzzy inexact processing.[1] The sciences of astronomy and physics have long used conceptual metaphor productively, in such terms as “event horizon”, “gravity well”, “curved space”, “time dilation”. The same technologies which provides the mechanisms for such term usage are available to robotic and geo-spatial processing.[2] Productive conceptual metaphors can be used functionally on computers.

Keywords: Second Order Metaprogramming, Conceptual Metaphor, Cognitive Fusion, Ontology

I. INTRODUCTION

Mobile robots operating outside the closed world of the factory face a much more complex working environment. Not only is there the open-world "other agents" problem to deal with but the objects and situations are more diverse and complex. Whereas precision and exactitude is desired in bridge building often it is not attainable in the unstructured natural environment. Being able to handle imperfect sensor data provides a mobile robot with an extra measure of coping. Having programming which governs the robot which does not use or require high-precision is also useful in adequately dealing with the unstructured natural environment. In the case of the DARPA Grand Challenge the programmers of the participating vehicles had some idea of the *kinds* of things they would need sense and deal with, objects, paths and depressions, for example. A robot system such as OPRS permits one to set up pre-programmed "notions" (representations in memory) for paths, way points, obstacles, for example and derive the presence of these by examining sensor data. Typically, the way points and obstacles may be treated as fixed things at a given location, which is not time varying. These constructs may be handled succinctly by means of relatively simple scalar representations of location, size and so on.

Planning the path of travel, a typical job of on board computing systems in an autonomous mobile robot, might be performed using the A* approach [3] [4] [5] [6]. The robot drive system itself requires specific instructions to operate with it cannot use inexact, waffle values. Approaches such as A* provide the system with point locations (exactitude) with which to operate. We see in this paper means of effectively fuzzifying approaches such as A*.

II. SITUATED

Deciding what to do and planning benefit from a tolerance of inexact reasoning because it brings a measure of flexibility. Point precision unrealistically demands exactness which often cannot be attained in the field where real robots operate. The tolerance of inexactness provides the ability of abstraction to greatly benefit the cleverness of the robot's planning function. Plans which substitute inexact values and inexact actions for precise ones give the plans a flexibility closer to that of biological systems, who never know precisely where their limbs are nor distances from themselves to things of interest in the world. This *situatedness* (relating both perceptions and measurements of the world [7] and reasoning relationships to oneself or substitute-other origin) is what provides biological systems with such reasoning power to cope with a complex world.

When we were babies daddy towered over us like a giant (metaphor, based on perceiver's height) and when we were in our later teens perhaps daddy was actually a bit shorter than us (same metaphor). This shows the "values" (giant, shorter) that metaphors "have" depending on the source of the metaphor ("my" height) can be quite different for the same objects (daddy).[8] Thus for this metaphor not only is "my" height the context for the metaphor but there is also a meta-context which is "my" age. ("My" is the perceiver, self). Context is which features and dimensions of all those which the system "knows" of are relevant, salient to the moment or task at hand.

Similarly when we were babies thrashing about in the crib we noticed the correlation between our reach (general arm movements for that matter) and what we saw with our eyes. That conceptual correlation provided us with a sense of space, of depth (3D volume, extent). I suspect that if we had no arms or legs and only eyes our sense of depth, if any,

would be dramatically curtailed. Measuring the distance from the laser range finder on a robot to the current location of the robot's hand does not provide a sense of depth. Many such measurements would have to be taken (with the hand at different locations obviously) and these would have to be combined with concomitant (robot limb) proprioceptive information (perhaps angles at inflection points and motor information as well).

For activities very close to the robot vehicle itself, such as an arm / end effector (hand) performing some task such as turning a valve on a pipeline, situated-ness is important. By providing a (vision) sensor coupled feedback of the arm position combined with arm-propriceptive feedback the robot would be able to gauge the location and orientation of the hand both by virtue of the proprioceptive feedback and by the vision sensor data interpretation of the same object. Thus a dual modality "sense" of "how far I can reach", "how big something is that I can pick up", etc can be obtained from vision and proprioception. The result is a visual sense of the scale (size) of things I (the robot) can manipulate and the (general maximum) distance I can reach. This latter knowledge allows the robot to figure out (by itself) that if something that it wants to pick up is farther away than its reach then it has to move its body, and move it so that the object falls within the maximum hand reach, not to mention that the body of the vehicle has to be oriented so that the arm can reach out and grasp the desired object. Notice that the metaphor involved in planning this does not require that the vehicle body end up aligned in the same way as it was when it detected the need for a position change. The metaphor for arm reach, say, is a half-circle where the flat part of the circle corresponds to the side of the vehicle and the center of the flat part of the circle is where the base of the arm is located. As long as that half-circle has the object to be picked up in it it really doesn't matter the orientation of the side of the vehicle. The great freedom or flexibility of this plan means that the actual operation of the vehicle based on this metaphorical plan does not try to maintain either super accuracy nor retain unnecessary states such as the orientation angle of the vehicle at the time it was determined that a change in vehicle location was required. Metaphor-based planning is potentially much more flexible than exact-match approaches.[9]

III.REPRESENT SALIENT ASPECTS

Metaphors are used to "explain" or model phenomena or even concepts. They are productive because they are inexact, imprecise and yet capture or *represent the salient aspects or features* of the phenomena or concepts. Conceptual metaphors are used in daily life by all of us, perhaps even without our recognition that they are metaphors, or analogs.

High-quality computer games use a lot of code to "process physics", like gravity, gas expansion, etc. and generate reasonable action / behaviour to make the games more like the real physical world. Autonomous mobile robots benefit

from the same processing capabilities as it helps to interpret sensor data and generate expectancies for what may happen [3] [4] [5] [6], such as sunshine and gravity causing rain, rivers, lakes, oceans, rain forests, jungles [10].

Metaphors of various types help reasoning by providing a known description or process to address reasoning about something which is not a known description or process. An example, the container metaphor is used to talk about the mind. The head contains the brain which contains the mind. There are expressions such as "He lost his mind", "He lost his head", "He didn't have a thought in his head" and, "Get that thought onto paper(!)". Words are containers for thoughts, which themselves are objects. Time is like money to be spent or wasted, or saved. These are all common metaphors that we think with daily and generally pay no heed to them as such.

It is the ability to functionally process inexact things which allows this very useful abstraction, the substitution of something *similar* or *corresponding* for unknowns. It allows reasoning about things that are not perfectly familiar with (/ by using) those that are, or at least felt to be. An apple falling from a tree becomes dropping a bomb from a UCAV. Throwing a spear becomes shooting a rifle. But, unfortunately, climbing a tree may seem a partial solution to getting to the moon. This is precisely because "climbing trees" is inappropriate activity in the domain of traveling between celestial bodies. (Jack and the Beanstock, Baron Munchausen). This also shows up one of the problems with modeling in general. For example, if I can sprint from a standing start to ten feet away in 3 seconds that means I have an acceleration of about 3 feet per second per second, let's say. In order for me to model my sprinting to the moon, by achieving escape velocity, all I need to do is divide escape velocity by 3 feet per second per second. That will then tell my model how many seconds I need to sprint from a standing start to achieve escape velocity and thereby travel to the moon. You and I realize how absurd this is but a computer, having no common sense, may escape noticing that. This simply goes to show that neither modeling nor abstracting (metaphors) can be performed willy-nilly but must be done in a principled manner in order that the result be functional. Part of the solution is to use relevancy / conditionals, to use relevant domains and dimensions / aspects. Partial ordering of the innards of the metaphor is the answer to this.

The sciences of astronomy and physics have long used conceptual metaphor productively, in such terms as "event horizon", "gravity well", "curved space", "time dilation". The same technologies which provides the mechanisms for such term usage is available to robotics / geospatial processing. Productive conceptual metaphors can be used functionally on computers. This may be done by employing ontologies which provide requisite background knowledge to the system and qualitative logic processing. For example, a metaphorical comparison between two geospatial entities is

qualitative and able to arrive at (useful) data when often strict-equality and simple scalar comparison either outright fails or returns a possibly nonsensical answer.

Just as "near" is a word (having a defined process which implements or calculates it) in computing with words there are statements which can use it in a natural language sense and be functional.

A sigmoidal function is used to map grade of membership for a range of distances in the function `IsNear`, which follows. `g1` is a named computational function shown in the ontology fragment:

```
<rdf:Property ID="IsNear">
  <rdfs:comment>has a degree of nearness (by value).
  g1(x)</rdfs:comment>
  <rdfs:range.rdfs:resource="#www.open-meta.com/2001/
  IsNear" />
  <rdfs:domain rdfs:resource="#SvgEntity" />
  </rdf:Property>
```

$IsNear = g1 = 1 - (1/2 + 1/\pi * \arctan(\sqrt{(x2-x1)^2 + (y2-y1)^2}) - k(1)/k(2))$ [10].

"`g1(x)`" from the comment field of the above ontology fragment shows that "`x`" is the argument for the function named "`g1`". In this case the argument, "`x`" is the value of the distance which separates two objects of focus. Each of these objects has a centroid and the distance separating the two centroids is called the `centroidsdistance`. Therefore the comment refers to the function and argument `g1(centroidsdistance)`. The `centroidsdistance` is calculated by:

$$centroidsdistance = \sqrt{(x2-x1)^2 + (y2-y1)^2}.$$

`k(1)` is the value of the center of the function curve, and `k(2)` is the value of the slope of the function curve (also known to some as its `Q`).

IV. FUZZY CALCULATIONS

This fuzzy calculation is performed using a context, that is what provides the parallel with human subjectivity in fuzzy function usage. The context for `IsNear` (above) is derived from the spatial domain to which the "near" calculation is being applied. For example, let's say that the spatial domain is an SVG picture which has extents 0, 0, 400, 300. Let us define "diagdistance" to be the length value of the diagonal from the origin (`x=0, y=0`) to (400, 300). We might use the `centroidsdistance` calculation (above) to compute this value. The context then, a multi-member structure, has "diagdistance" = `centroidsdistance (400, 0, 300, 0)`. "Diagdistance" may be used as a reference to "scale" fuzzy measurements from the SVG picture. For example, in the overall scenario of the SVG picture above, "diagdistance"

may be used to provide a scale for the x-axis of the `IsNear` calculation, scaling the sigmoidal function. That then provides a perspective for the `centroidsdistance` value. There is nothing special about SVG here, any two-dimensional spatial (named) frame of reference could be used for the domain depiction. Other context elements that are obtained for the context structure from the spatial domain to which the "near" calculation is being applied are "xwidth" and "yheight", in this case "400" and "300" respectively. In the case where the context of the processing was to fuzzily label some width-distance (x-axis distance) item in the spatial domain being processed such as the one above then the value of the width-distance would be compared with "xwidth" and a hedged name computed for the result, in the form of "hedge-name wide".

V. STRUCTURAL METAPHOR

In the case of "The squirrel ate the acorn near the foot of the tree.", we have fuzzy and metaphorical elements in the statement. "Near" is a fuzzy term that is in context and "foot" is a metaphor whereas "roots" would not be. The context for "near" is "tree", that is, the location (and morphology) of the tree is considered the "(axial) origin" or "zero point" for the location on which the calculation `IsNear` is made. The morphology of the tree is taken into account such that the "location" on the tree-morphology where the distance calculations are made is "at" the "location" equivalent to "foot". Here "foot" (of the tree) is a metaphor, trees do not have actual feet and a search of a database on tree information would not find any information on "feet" there. The metaphor here is a structural metaphor, which can be seen by looking at a drawing or even a photograph of a person standing. The pixels and lines in the picture or drawing which constitute the / a foot are indicated by a human. The computer is able to determine that the "foot" so indicated is always at the bottom of the object in the picture or drawing and constitutes a certain average percentage of the overall picture or drawing by area, it has morphology which is captured. As a result the computer is able to use fuzzy calculations, such as "near" and "far", "morphological extent and characteristic" to show that "foot" is always "at" the "bottom" of the object depicted. Hence, metaphorically speaking, the "foot" of a tree (object) is located "at" the "bottom" of the tree. (The roots of a tree may be found in a database lookup, both location and perhaps as a visual or drawing, and hence "known" to be "at the bottom" of the tree.) Since the roots of a tree are actually at the bottom of a tree "roots" is not metaphorical it is literal. The basis for the context which provides a scale for "near" in this case is "tree", specifically the comparison of the general distance from the (actual or estimated) center of the tree trunk to the (general or average, or even believed to be) outside edge of that tree, with some actual (or estimated) distance of the acorn from the (edge of) the tree. If the acorn was about 5 feet away from the tree and the tree seemed to be about 3

inches in diameter then the fuzzy calculation would return a value of "not very near", whereas for a twenty foot diameter redwood the value would be (something like) "quite near".



(1) Qrio robot (Sony), an actual bi-pedal autonomous robot.

The picture (1) is 200 by 327 elements in size with 0,0 (the origin) located in the upper left corner. Notice that a rectangular-area around the (robot's) feet is located by the two points (x,y) (58,75) (153,125). (This box is not shown explicitly in the picture.) Assuming that the y-axis at 327 is labeled "bottom", if one applies the "near" fuzzy function with a function origin at y=327 along the Y-axis of the picture then one finds the box to be "rather 'near'" the "bottom" (spatial location). This is how the fuzzy robot system is able to functionally process structural metaphors, such as "foot of tree," "foot of cliff." [11] (correspondence) That being said we now have a name which can be used in ontologies as a linguistically derived term. When we look at the CYC knowledge-base [12] we see that it has defined a number of valuable useful knowledge items to do with foot. A piece of the CYC knowledge-base is shown next. This OWL (Web Ontology Language) item is just one of a myriad of OWL entries in the CYC knowledge-base. CYC may be used to infer tacit information, that is information which is not explicitly present in an input / data set. Notice that the CYC OWL item below tells the computer in semantic web terms about "foot".

```
<owl:Class rdf:ID="Foot-AnimalBodyPart">
<rdfs:label xml:lang="en">feet (types of things)</rdfs:label>
<rdfs:comment>The collection of all vertebrates' feet.
A foot is a terminal part of a #SVertebrate #S$Leg. Feet are
used in locomotion, support, balance, kicking, etc.
</rdfs:comment>
<guid>bd58be93-9c29-11b1-9dad-c379636f7270</guid>
```

```
<rdf:type rdf:resource="#PublicConstant"/>
<rdf:type rdf:resource="#SymmetricAnatomicalPartType"/>
<rdf:type rdf:resource="#AnimalBodyPartType"/>
<rdfs:subClassOf rdf:resource="#Appendage-
AnimalBodyPart"/>
<rdfs:subClassOf rdf:resource="#Individual"/>
<owl:disjointWith rdf:resource="#Digit-AnatomicalPart"/>
<owl:disjointWith rdf:resource="#Limb-AnimalBodyPart"/>
</owl:Class>
```

This ontological item tells the computer that a foot is not a finger, arm, or leg; and that its location is a terminal part of a leg (i.e. at the end of a leg, not the middle or elsewhere). This ontological semantic knowledge is valuable for it allows the program we are discussing in this paper to identify / locate a foot in a visual data set yet without being committed to a particular photograph of one, nor drawing or X3D three-dimensional data set of a foot.

VI.RELEVANT TRANSFER

Saying that a person "has a head the size of the Great Pumpkin" is metaphorical, and fuzzy, and the computer can perform logic / calculations to "compute" the meaning of the statement: context, usuality. "Fuzzy calculations" which in part constitute computing with words are also used to constitute the meaning of metaphors (in a graded rather than solely symbolic way) so as to make it possible to remove them from being simply items in an enumerated semantic set.

An example of a metaphorical plan that might be used by a robot would be an "easter egg hunt" (plan). While sounding frivolous it is in fact a good example because there is *relevant transfer* of both goals and actions from the *activity* of an "easter egg hunt" to that (activity) of locating and acquiring dispersed objects of value at unknown and possibly hidden locations. An autonomous mobile robot might be "coached" (sort of like "given a clue") by saying that the goal is to treat the sought after objects as though easter eggs and get them. This "hint" allows the robot to detect the relevance of using an "easter egg hunt" metaphorical plan. Try to explain such a complex activity to a robot using Cobol.

Another metaphorical plan "Don't get too close to a deep hole." "Near" (a synonym of "close") has been explained as a computation, "too close" is a hedged version of "too near" (in this case "too" is treated as a synonym for "very"). A "hole" is computationally treated as a depression in a surface. "Deep", computationally, is treated as though "very distance ratio (yyvalue / xxwidth)." In English, the calculation looks at the ratio of the y-axis direction of the depression versus the magnitude of the diameter of the depression, and if the hole is substantially deeper (larger Y-axis value) than it is wide it is called a "deep hole". It has a grade of membership which varies with that ratio. Using a measure or sense of the diameter of the depression as a scale (basis) for distance, calculate what actual distance

corresponds to the hedged distance term "very near" and then define a radial around the center of the circle whose radius is at the "actual distance" and keep the robot trajectory farther away from the center of the depression than that. "Don't get too close to a deep hole." might be considered as advice, a plan set to modify or generate a current plan of activity.

Earlier we saw another fragment from the DAXSVG ontology, that fragment was the term "IsNear". There we saw the fuzzy calculation of the grade of membership for the function "near", for two point-objects. The DAXSVG ontology fragment "Above" is shown next. It is followed by a Java-based implementation of the logic used to determine the correct meaning or semantics of the term "Above" in the context of the two objects whose location is being examined. Instead of returning just "true" or "false" a grade of membership is returned by the computation which signifies the fuzzy degree of Above-ness of one object from the other.

```
<rdf:Property ID="Above">
  <rdfs:comment>centroid of OBJY1 is anywhere above the
  centroid of OBJY2. Above means a y-axis value for OBJY1
  which is less than the y-axis for OBJY2. (2D)
</rdfs:comment>
  <rdfs:range rdf:resource="#SvgEntity" />
  <rdfs:domain rdf:resource="#SvgEntity" />
</rdf:Property>
```

```
public real Above( int y1, int y2 )
{
  if (y1 < y2)    //2D Above
  {
    /* Far() = 1 - IsNear() */
    return 1 - ( IsNear(y2-y1) );
  }
  else
  {
    return 0;
  }
}
```

In a (2D) frame of reference (corners 0, 0 200, 300) the calculation (of Far (which is the fuzzy complement of Near) peaks at 300 y-axis units of object separation. That is to say that the grade of membership of the Far calculation is 1.0 when the separation of the two objects is 300 y-axis units. What the calculation is doing is obtaining the grade of membership of the Far function (1 - IsNear) in the "Above" context. So, if object OBJY1 was near the top of the frame of reference (close to y-axis = 0) and OBJY2 was near the bottom of the frame (close to y-axis = 300) then a hedged fuzzy natural language rendition of the separation (of OBJY1, OBJY2) would be "very Far Above". Notice that this kind of system can "see" the equivalence of "not very Far Above" and "very Near Above". ("Not" is complement.)

Object1 (y-axis value of y1) is Above Object2. (x,y = 0,0 is at the upper left corner in this axis system.) Each of the RDF predicates in DAXSVG has a program member written in Java. These programs, like the one above, provide required actions for the DAXSVG ontology. The Java program above is purposely simplified for illustrative purposes. It does not show the code used to examine the arity of the dimensionality of the frame where the comparands, supposedly one above the other, occur, and the concomitant selection of the Y- or Z-axis for testing of relative location of those comparands.

Below are several items from the CYC ontology [12] which convey the meaning of various aspects of "above" (a synonym of "before"). Remember that there is nobody home in a computer and so this pointy bracket verbiage is necessary to provide a modicum of enlightenment to the computer, of stuff which we tacitly know and take for granted everyday.

Looking at the comments in these CYC ontological items you may see that the natural language terms for fuzzy predicates, such as ABOVE, BELOW, Directly, Generally, Up, Higher occur there. Computing with words technology can provide real computations employing these concepts / predicates, as illustrated above with the fuzzy term "IsNear". (Metaphors like "foot of.." use the same computing principles as these. By detecting and applying relevant contexts, such as which dimensions make sense to compare and process, these computations can bridge fuzzy systems / computing with words with ontological technologies. These contexts provide a measure / identity of the "partial" aspects of "partially structured metaphors.")

```
<rdf:Property rdf:ID="above-Directly">
  <rdfs:label xml:lang="en">above - directly</rdfs:label>
  <rdfs:comment>(#$above-Directly ABOVE BELOW)
  means either that (1) the volumetric center of ABOVE is
  directly above some point of BELOW, if ABOVE is smaller
  than BELOW; or that (2) some point of ABOVE is directly
  above the volumetric center of BELOW, if ABOVE is larger
  than, or equal in size to, BELOW.</rdfs:comment>
  <guid>bd58fbde-9c29-11b1-9dad-c379636f7270</guid>
  <rdfs:subPropertyOf rdf:resource="#above-Generally"/>
  <rdfs:domain rdf:resource="#SpatialThing-Localized"/>
  <rdfs:range rdf:resource="#SpatialThing-Localized"/>
</rdf:Property>
<rdf:Property rdf:ID="above-Generally">
  <rdfs:label xml:lang="en">above</rdfs:label>
  <rdfs:comment>(#$above-Generally OBJ1 OBJ2) means
  that the #SpatialThing-Localized OBJ1 is more or less
  above the #SpatialThing-Localized OBJ2. To be more
  precise: if OBJ1 is within a cone-shaped set of vectors within
  about 45 degrees of #Up-Directly pointing up from OBJ2
  (see #Up-Generally), then (#$above-Generally OBJ1
  OBJ2) holds. This is a more general predicate than #Above-
  Directly (q.v.), but it is a more specialized predicate than
  #Above-Higher (q.v.). It probably most closely conforms to
  the English word above.</rdfs:comment>
```

```

<guid>be69c623-9c29-11b1-9dad-c379636f7270</guid>
<rdfs:subPropertyOf rdf:resource="#above-Higher"/>
<rdfs:domain rdf:resource="#SpatialThing-Localized"/>
<rdfs:range rdf:resource="#SpatialThing-Localized"/>
</rdf:Property>
<rdf:Property rdf:ID="above-Higher">
  <rdfs:label xml:lang="en">above - higher</rdfs:label>
  <rdfs:comment>(#$above-Higher OBJ-A OBJ-B) means
that OBJ-A is `higher up&apos;&apos; than OBJ-B. Since
most contexts are terrestrial (see
#$TerrestrialFrameOfReferenceMt) `higher
up&apos;&apos; typically means that the
#$altitudeAboveGround of OBJ-A is greater than that of
OBJB.</rdfs:comment>
  <guid>bf020f6c-9c29-11b1-9dad-c379636f7270</guid>
  <rdfs:subPropertyOf rdf:resource="#spatiallyDisjoint"/>
  <rdfs:domain rdf:resource="#SpatialThing-Localized"/>
  <rdfs:range rdf:resource="#SpatialThing-Localized"/>
</rdf:Property>
<rdf:Property rdf:ID="above-Touching">
  <rdfs:label xml:lang="en">above - touching</rdfs:label>
  <rdfs:comment>(#$above-Touching ABOVE BELOW)
means that ABOVE is located over BELOW and they are
touching. More precisely, it implies both (#$above-Directly
ABOVE BELOW) and that ABOVE #$touches BELOW.
Examples: a person sitting on a chair; coffee in a cup; a boat
on water; a hat on a head. (Note that not every point of
ABOVE must be higher than every point of
BELOW.)</rdfs:comment>
  <guid>bd58f620-9c29-11b1-9dad-c379636f7270</guid>
  <rdfs:subPropertyOf rdf:resource="#above-Directly"/>
  <rdfs:subPropertyOf rdf:resource="#touches"/>
  <rdfs:domain rdf:resource="#PartiallyTangible"/>
  <rdfs:range rdf:resource="#PartiallyTangible"/>
</rdf:Property>

```

VII.FUZZY ACTIONS

An SVG program can implement a (stepped) radial-gradient. The concentric areas of isometric values are used as a metaphor for grade of membership values vis-a-vis spatial location with the exact circle center having highest value. It is a graphic of a step-function which maps distance with grade of membership. The way it is used in a metaphorical robotic action is this: the center of the circle is set to the closest point available corresponding with the centroid of an object which is to have another object placed upon it. This metaphoric "object-placement" function [13] then provides a single point (on a surface) with which to locate the centroid of the object being placed.

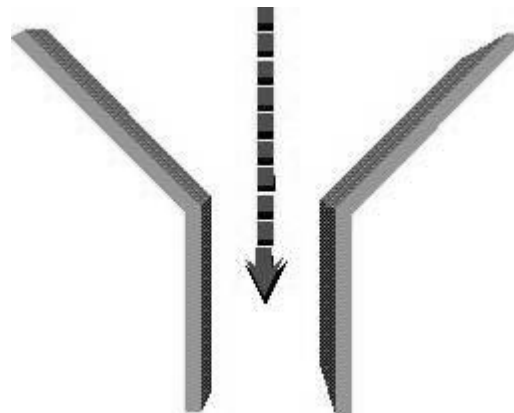
The set of points that occur within the inner most circle might be hedged "very good", the next most inner ring area "rather good", the next ring "fair" (fairly good), and the outermost ring "not good" (fuzzy complement of "good"). So a fuzzy robot using this metaphor for location to place objects during an "on top of" type activity could use an advice like "use good enough location". This means that

placement activity of such a robot, rather than being precisiated compulsion of exactly closest to the receiver's centroid, the fuzzy robot may be more like a human and , not knowing exactly where the centroid is on the receiver's surface, places the object in a reasonable but inexact location. Each time the human does that the centroid is likely placed at a different location [14], and that's okay. This metaphoric approach does much the same thing. The location is fuzzy, and the placement also avoids being "not good" at the same time. It also works if the centroid of the radial-gradient used for placing is placed in a fuzzy-area zone centered on the receiver "generally near enough" to where the actual centroid is in the receiving object. When we place things "by eye" this is what we do, we never measure exactly where the centroid of things is to place objects.

VIII.DE BONO DIAGRAMS METAPHOR

The next illustration (2) is an example of how Edward de Bono uses pictures in a book [15] to explain management processes or recommendations of thinking and action types for management. The details of the meaning of the pictures / diagrams is best found by reading his book "Atlas of Management Thinking." His illustrations are all snapshots not video clips and are designed to be suggestive of certain constructs, in particular non-permeable diagram members, motion vectors as suggested by various kinds of arrows, "motion" (change of location) along a path and a few other basic, reoccurring conceptual primitives.

The present author has written SVG SMIL-based animation programs which draw these non-permeable

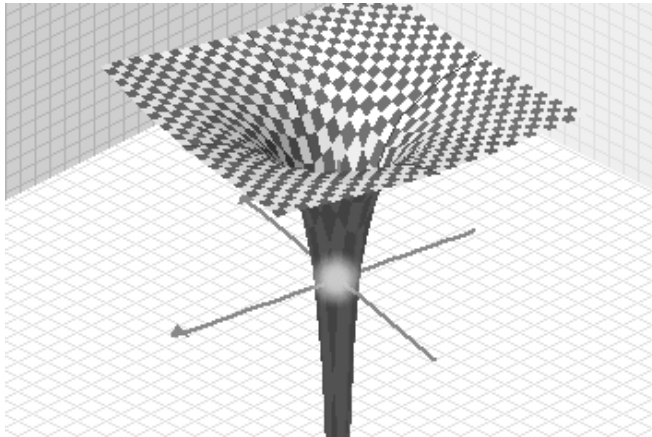


(2)

diagram members and animates arrows to follow paths as suggested in de Bono's illustrations. SVG is a graphics language which uses XML as a representation, not proprietary binary (opaque) stuff like in Flash animations. The upshot is that the SVG programs can be parsed with an XML parser and so semantic-web type technologies may examine SVG programs directly and discover semantic knowledge therein. Through such parsing and discovery the actual SVG program can be translated into fuzzy

representations so as to allow them to be processed via computing with words technologies. For example the illustration (4) has marks in it which to our eyes form or constitute an arrow, which has an implied (by the dashes) pathway of “travel” or progression. SVG animation can easily draw an arrow moving along the path. Various aspects of the illustration can be made usable by a fuzzy system. As was shown earlier in the paper, the dimensions of the picture used as a relative size calibration, the dimensions of the non-permeable diagram members (the “Y” lines in this case, and the (implied) motion of the arrowhead. The SVG animation, the fuzzy mapping of it, and the semantic discovery components can be taken as a whole to provide a computer based “understanding” of what de Bono was trying to teach as “algorithms for thinking” (for management persons). The illustration (4) is thus like a snap-shot of a motion clip, it is an analog or analogy, a metaphor of action or behaviour and metaphoric / fuzzy / computing with words technologies can be used to capture and functionally represent these.

IX. PLANNING WITH CONCEPTUAL METAPHORS



(3)

Illustration (3) is a graphic of a gravity well, the extrusion in the gravity plane produced by a mass. The morphology of the extrusion is governed by mathematics which calculates Einstein's field equations [16].

By generating animations of the table showing the “behaviour” of animated “balls rolling on the surface” metaphors of the terms “gravity well”, “event horizon”, “black hole” and so on are made. A metaphoric system can, then, simulate the launching of various (spherical) masses onto the model surface at various locations and speeds and “observe” (data record) the results. A hypothesis forming / “discovery” system could take the data and generate theories. Metaphors which engender the analogy of extruded plane can then be used to (computationally) “theorize” with and to learn with. (Years ago (1986) “Boyle's Law” was rediscovered by a discovery program (BACON) [17] developed by Pat Langley. [18])

An SVG animation which the author wrote to demonstrate an XML (SVG) means of capturing metaphorical and

analogical knowledge and providing visual / diagrams to represent such, is at open-meta.com, called anim02.svg.

SVG may have embedded meta-data, providing (functional) ontological and fuzzy information inside the XML picture itself.[19] An analog or metaphorical “sense” of “value” or “scale” can be conveyed in such animations by carefully deciding what settings to use for SVG's animate elements of “time-speed”, defined by the parameters “dur”, “from,” and “to”.

X. References

- [1] D. Dodds, 'Fuzziness in Knowledge-Based Robotics Systems,' 1988. *Fuzzy Sets and Systems* 26: North-Holland 179-193.
- [2] D. Dodds, 'Fuzzy Logic Computer Implementation of Metaphor from Ordinary Language,' AAAS Annual Meeting (American Association for the Advancement of Science) 1981.
- [3] D. Bourg, G. Seemann, *AI for Game Developers* Cambridge, Mass: O'Reilly Media, 2004.
- [4] S. Rabin Ed., *AI Game Programming Wisdom* Hingham, Mass: Charles River Media Inc, 2002.
- [5] S. Rabin Ed., *AI Game Programming Wisdom 2* Hingham, Mass: Charles River Media Inc, 2004.
- [6] T. Jones, *AI Application Programming* Hingham, Mass: Charles River Media Inc, 2003.
- [7] L. A. Zadeh, “From Computing With Numbers To Computing With Words — From Manipulation Of Measurements To Manipulation Of Perceptions,” *Int. J. Appl. Math. Comput. Sci.*, 2002, Vol.12, No.3, 307–324.
- [8] G. Lakoff, M. Johnson, *Metaphors We Live By* Chicago, Ill: The University of Chicago Press, 1980.
- [9] D. Dodds, 'Fuzziness in Knowledge-Based Robotics Systems,' 1989. *Mobile Robots IV: Volume 1195 SPIE - The International Society for Optical Engineering* 56-65.
- [10] D. Dodds, “Qualitative Geospatial Processing, Ontology and Spatial Metaphor,” presented at the GML And Geo-Spatial Web Services Conference 2005, Vancouver, BC, July 18–22, 2005.
- [11] D. Gentner, K. Holyoak, B. Kokinov, *The Analogical Mind* Cambridge, Mass: The MIT Press, 2001.
- [12] OpenCYC knowledge base reference. Available: <http://www.opencyc.org/downloads>
- [13] D. Dodds, 'Natural Language Processing and Diagrams,' The 2004 International Conference on Machine Learning; Models, Technologies and Applications.
- [14] If the mass of the object is not uniformly distributed, we can characterize the mass distribution by a [function](#) $w(x)$ which indicates that the weight is some function of distance x from a reference line. If we can determine the form of the function, there are methods to perform a calculus integration of the equation. We will use the symbols “ $\int dx$ ” to denote the integration of a continuous function. Then the center of gravity can be determined from: $cg = \frac{\int [x * w(x)]dx}{\int [w(x)]dx}$. Available: <http://exploration.grc.nasa.gov/education/rocket/cg.html>
- [15] E. de Bono, *Atlas of Management Thinking* Bungay, Suffolk: Chaucer Press, 1981.
- [16] Gravity From Wikipedia, Einstein field equations.
- [17] P. Langley, H. A. Simon, G. L. Bradshaw, J. M. Zytkow, *Discovery program (BACON), Scientific Discovery: Computational Explorations of the Creative Processes*, Cambridge, Mass: MIT Press, 1987.
- [18] Animated Boyle's Law illustration. Available: <http://www.grc.nasa.gov/WWW/K-12/airplane/aboyle.html>
- [19] D. Dodds, *Professional XML Meta Data* Wrox Press Inc; 2001.