Zygmunt Ryznar
dr emeritus Cracow Poland
zygmunt.ryznar@gmail.com

Linguistic presentation of objects

(types,structures,relations)

Abstract

Paper presents phrases for object specification in terms of structure, relations and dynamics. An applied notation provides a very concise (less words more contents) description of object.

Keywords
object relations, role, object type, object dynamics, geometric presentation

Introduction

This paper is based on OSL (Object Specification Language) [3] dedicated to present the various structures of objects, their relations and dynamics (events, actions and processes). Jay W.Forrester author of fundamental work "Industrial Dynamics"[4] considers the importance of relations: "the structure of interconnections and the interactions are often far more important than the parts of system".

Notation

<!...>	comment
< >	container of (phrase,name...)
≡>	link to something external (outside of definition))
<def > </def>	start-end of definition
<spec> </spec>	start-end of specification
<beg> <end>	start-end of section
[..] or {..} [1]	list of assigned keywords
:[or **:**{	structure
^<name>	optional item
(..)	list of items
xxxx(..)	name of list
=	value assignment
@	mark of special attribute,feature,property
@dark	unknown, to obtain, to discover
::	belongs to
:	equivalent name (e.g.shortname)
#	number of
|name|	executive/operational object
ppppXxxx	name of item Xxxx with prefix 'pppp '
XXXX	basic object
UUUU.xxxx	xxxx object belonged to UUUU object class
& /	conjunctions 'and' 'or'

1 If using Latex editor we suggest [..] brackets instead of {..}.

<->[2] or ↔ Bidirectional unitary relation 1:1
<- or ← Back directional unitary relation 1:1
-> or → Forward directional unitary relation 1:1
<=> <= => Many to many relations
<i=> <i= =i> Intrusive relations (making a compulsory change)
.. More
… Much more

Object types

object.type{[eObject<!elementary atomic object >,
dObject<!dynamic object >,
iObject<!informational object >,
vObject<!virtual object >,
sObject<!smart object)>
sObject[noiceReduction,selfTeach,selfRepair,selfKill,selfRestore,selfRestart],
oObject<!open object>,
oObject[tuning(input(parameters,data),output(info,messages),
structure(addComponent,addRelations))]
incObject<!incarnation of living object>,
copyObject<!copy of object>
binObject:Binder<!collection of objects having common task>,
taskObject:Task<!task to be performed>,
interObject<object being parts of several objects - intersection of objects>,
capsObject:Capsule<!portion of items- reserved for a given user, purpose etc.>]
involvedObject[(list of objects) when <condition>]}

Object dynamics

event:ev<!-elementary atomic fact >
operation:op
action:ac<!sequence of operations or events>
process:pr<!sequence of actions and events>
pr[trigger,<actions><events>,close-down:endEvent]
dynamics[event,operation/transaction,action,process]
dynamics[ev,op/tr,ac,pr]<!short notation>
dynamics.scenario[evSc,opSc,acSc,prSc]<!event scenario,….>
reverseMode[rev,rAc,rOp,rTr]<!back to the previous state>
scenario:sc<!predicted sequence of actions and events>
scenario.rank[best,middle,worst]
object.Info<!information visible at the moment of access>
keywords:kwords<!additional keywords in def >
olh<!object life history>[timeline,events,aging-curve]

Object interactions-relations

Types of relations:
- dynamic (thru processes, actions, events)

2 If typing on keyboard

- structural (whole, component)
- active(intrusive,accepted),passive

relations[role,relation,relator]
- role
  role[interface,integrator,component,monitor,commander,
       driver,trigger/activator,reactor,agent,executor,generator
       locator,executor/performer,initiator,terminator,destructor,
       participator,owner,stockholder,customer,supplier;partner,employee]

 relation[activated by,activates,assisted by,built from ,
            appearence depends on ,belongs to/is owned by ,
            exists as satellite of <object>,calls <object> (<interface>),
            consists of <parts>,contained in/contains,
            controlled by/controls,derived from,
            existence depends on,exists when/in/for,
            included in,linked to ...by/links,
            refers to,relates to,related by affinity,
            represented by/represents,involved in,
            shared by/shares,used by/uses]
relator <!maker of relation>
        relator(marriage,employment,…..)
relation:[<item1><phrase><item2><!item1 may be assumed as defined before>
relation{[singular,iterative(standard,recursive)],[sequential,parallel(many-data_sources),mixed]}
          <! if not specified – singular and sequential are assumed>
           <!iterative but not necessarily approximative
           <!iterative normativ – result of each iteration is taken>
           <!iterative approximativ – final result is taken - for example when teaching of neural
              network>
 state[active,inactive,dark,dormant,suspended,aborted,
       variable,invariable,idle/waiting,lost,expected,deleted,homeless]
 status[generic,real,virtual,undef ined]
 reactor[acceptance,rejection,constructor]
 rank[critical,necessary,most wanted,optional,worst,best]
 rule[decision-table,when-if,formula].

Control-flow of objects

   ac(ev1,ev2,ev3, ..)<!action-sequence of events>
   pr(ac1,ac2,ac3,...)<!process>
   s(ev1,ev2,ev3, ..)<!sequential flow of events>
   p(ev1,ev2,ev3, ..)<!parallel flow of events>
   pr(s(ac1,s(ev1,ev2,ev3),ac2(p(ev4,ev5,ev6),(ev7,ev8,..))<!mixed flow>
   repetition[iteration,spring,spiral]
   activated by <..> with <initial-value> at <time-point>
              when <condition>
   finished at < > with <...> when <..>

Geometric presentation of objects

Generally, OSL does not concentrate on data structure and formal documents but on descriptive statements, but sometimes wording specification would be better expressed in a conceptual (non-graphic) geometric way focused on the layout and behaviour of objects.

A geometric view has two profiles. Profile 1 expresses standard OSL geometric view and consists of many conceptual geometrical figures. Profile2 includes flat figures like flowcharts showing input-output and function or program (in computer job sequence) and diagrams presenting the structure (e.g. class diagram). These profile2 items are of classic type and can be easily made manually or generated from detail specification.

A geometric view could be an inspiration for new type of data in the cognitive computing or bigdata mining, as geometric shapes may be discovered by analysis of various massive information. We assumed that geometric interpretation might be be more accurate and more evocative because it shows imaginably (particularly in case of spiral) the nature of objects at least in terms of space and time.

There are many geometric figures. Some of them are uncommon (swarm, bunch, blackhole, wormhole, freespace) and become a challenge to be taken up in conceptual modeling.

1. Spiral differs from classic iteration in that sense, that every scroll can possess different "engine" and content. Quite "powerful" are multistrand spirals. Examples of strands in business could be profit, competitiveness, credit ability etc. Such spiral in a business may illustrate for example the bankruptcy of company which falls into spiral that pilots call the "death spiral". A special type of spiral is an irregular hyperbolic spiral intersecting an asymptote infinitely many times.
   Spiral approach is known in software development where the spiral has four phases (planning, risk analysis, engineering, evaluation) and project repeatedly passes through these phases in every iteration.
2. Swarm relates to the homogeneous relocatable dense population. The path of moving swarm is a good illustration of global expansion of business or moving a business from one country to another.
3. Free space means unstructured or with no rules of structuring.
4. Tunnel is three-dimensional population with the value X (e.g. the sum of credits) along the time Y and densed acording to the third factor Z (e.g. credit ability).
5. Cylinder differs from the tunnel with the feature that it carries the values on the surface while the tunnel keeps them in the interior.
6. Triangle coud represent "shadows" of object on its 3 angles surface (a-axis, b-axis, c-axis), e.g. for employees population angles are age, education, sex. The father of a triadic approach was famous philosopher G.W.F.Hegel (1770-1831) who used triangle to visualize a 'system of science' as a triangle with angles: logic, nature, spirit.[5]
7. Blackhole marks the irreversible disappearance of the object (e.g. the bankruptcy of the company) and shows the „strength of drawing in" (e.g. a speed of the bankruptcy).
8. Darkbox is a place for a dark (or hidden) information.
9. Cloud means an external container equipped with secure gate to enter it.
10. Neural networks are networks of interconnected layers and nodes, which process information as a response to external inputs using methods unavailable in traditional calculations. Very known applications of neural networks are character recognition, human face recognition and signature verification. In business they are used when

solution is not based on a predetermined and preweighted criteria but on the past experience (e.g. in banking area - failures in loan granting, forecast time dependent

layout[shape(gProfile1),gProfile2,sparcity,density,variability]

geometric profile1

[free-space,swarm,bunch,network,neural-network,hierachy,line,triangle,tunnel,
curve,spiral,spring,circle,elipse,cylinder,sphere,ellipsoid,con,doublecon,prism,
fractal,fractal networks,squarepiramid,container,blackhole,wormhole,cloud,darkbox]

spiral
spiral[single-strand,multi-strand]
spiral[helix,logarithmic,hyperbolic,polygonal,rational,golden,
spherical,conical,circle-involute,cornu,daisy,epispiral,
archimedian,fermat,nielsen,ulam,poinsot,phyllotaxis]
helix<!a curve for which the tangent makes
a constant angle with a fixed line>
spiral.parameters[center-point,number-of-rotation,
number-of-band,starting-radius,points-per-rotation,
growth-per-rotation(horizontal,vertical)]

spring<!simple iteration>
swarm<!moveable homogeneous population with variable density>
bunch<!nonmoveable homogeneous population>
circle[edgeCircle<!population on the edge>,
insCircle<!population on the surface>]
elipse[surElipse, edgeElipse]
cylinder[edgeCylinder,insCylinder:tunnel]
edgeCylinder<!population on the edge of Cylinder>
insCylinder<!population inside of cylinder>
triangle<!ayout def ined by 3 factors always existed and related together>
container<!trunk, regular 3-dimensional figure)
blackhole<!"off the face of the surface">
wormhole<!place injected with foreign/strange information>
free space<!no limits, no predef ined structure)
line[single,multiline]
curve[parabola,hyperbola,….]
point<!something that may exist only as a single event e.g. big bang>
objPoint<!single event for a given object e.g. birthday>
solids[sphere,cone,pyramid,cube,cylinder]
polygons[rectangle,square,pentagon,hexagon,octagon]

neural-network
neural-network.category[biological,artificial]
neural-network.type[singlelayer,multilayer,Kohonen,Hopfield,convolutional]
neural-network.parameters[layers(input,intermediate,output)(hidden/visible),
connection-between-layers(backpropagation,...),variables,expected-values,
weights,rules-for-modifying-weights,learning-method]

geometric profile2

[flowchart(prFlowchart<!program functions>,jobFlowchart<!sequence of programs>),
diagram(<flat structure diagrams>)]

Conclusion

This paper presents a phrases that seem to be a good base for so called free structured object modeling that is is a collection of methods, techniques and tools for modeling the variable structure as an opposite to the widely used "well-structured" approach focused on top-down hierarchical decomposition.

References

articles:
[1] author = Ryznar, Z.
title = A conceptual model of an interfunctional data base system,
journal = Information and Management,
year = 1978, volume = 2, pages = 67–74,
[2] author = Ryznar, Z.
title = S&DL – Structured Design Language,
journal = Angewandte Informatik-Applied Informatics,
year = 1981, volume = 12, pages = 526--533,
[3] author = Ryznar, Z.
title = OSL Object Specification Language,
journal = Journal of American Academic Research JAAR,
year = 2017, volume = 5, pages = 47–52,
[4] author = Forrester J.W.
title = Industrial Dynamics,
address = USA,publisher = The MIT Press, year = 1961,
[5] www.hegel.net/en/e0.htm

APPENDIX - an example of definition

It could be applied a simplified concised notation for definition of complex object.

<def **BRAIN>**
brainId(personId &/biologicalId,type)
mainparts[forebrain(cerebrum(hemisphere,thalamus,hypothalamus),midbrain,hindbrain]
area[lobe,cortical-region]
lobe[frontal,parietal,limbic,occipital,temporal]
cortical-region[primary-visual,entorhinal,inferior-temporal,orbitofrontal,
lateral-prefrontal,inferior-parietal]
cortex[visual,sensory,auditory,..]
parts[cerebral-cortex,basal,dienceph,brainstem,cerebellum,hippocampus,spinal-cord]
brain-function[sensory(vision,hearing,smell,touch,...),
mental(association,speech,language-comprehension,coordination,...),
motor(eye-movement,voluntary-movement, ..)]
detail-parts[neuron,synapse,receptor,unpaired-electron,neurotransmitter,..]
neuron{[nucleous(mithochondria,membrane,cytoplasm,vesicle,perycarion),
myelin-sheath,schwann-cell,axon,dendrite],
form(multipolar,bipolar,unipolar)]}

view[active/passive (biological,chemical,energic,geometric,physical,medical,
logical,semantic,psychological,mathematical,ontological)]
<!active view is based on brain neural-network>
engine[thinking,emotions,info-retrieval,memorizing,intuition(trust,love,hate),
communication,total,...…]

emotionEngine => psychologicalView
emotionEngine[Lob.frontal(electrode-placements/neural network-area)]
psychological*View[love,hate,satisfaction,frustration,agression,enjoyment,anger,*
fear of insupport,regression,inferiority,persecution] </def>
totalEngine => medView
totalEngine(brainArea/neural network-area,disease-pattern)
medView(disease(neurodegenerative,neurological,..),injury)
neurodegenerative(dementia/alzheimers,parkinsons,huntingtons,..)
neurological(autism-spectrum,tumor,migraine,multiplesclerosis,epilepsy,stroke,..)
totalEngine =>/ → ontologicalView
totalEngine(brainStructure,brainContents)
ontologicalView(brainId,brainAging(neurons(dead,born)),diseasesHistory,
brainVolume(curve,...),brainUsage)
totalEngine =>/ → energeticView
totalEngine(brainStructure,brainContents)
energeticView(brainId,power-consumption vs aging,power-supply-disturbances)

</def brain>