

Zygmunt Ryznar

BSL - Business Specification Language

Abstract

Paper presents the way of business formal description that might be a contribution to business modeling techniques in terms of structure, properties and behavior of objects. A detailed banking example is given.

Keywords

business modeling, business specification language BSL, geometric view

Introduction

A business specification is written in BSL (Business Specification Language) that is the subset of OSL (Object Specification Language).

This language covers not only main objects (subjects) but also whole environment all around defined globally (at kernel) and locally (at subject level). There are no borders of definition. One could imagine description of universe or more (let's call it "galaxy") even. OSL includes also open objects which could be incorporated in any "place" like free electrons of atom.

Another feature of OSL is an object orientation. Objects may be concrete (physically existent) and abstract (conceptual, virtual). The physical objects are viewed ontologically. Any object has at least its own name, identifier, properties, structure, interface and relations to other objects, own behaviour and "life-history".

OSL does not concentrate on the data structure and document formats but on descriptive wording specification (even in case of geometric figures).

Geometric view

A geometric view is a quite new feature in specifications languages. It 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 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 according 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 could 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.[19]
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 variables such as net income for each month of a year. Neural networks are applied also in finding trends in large quantities of data (currency, stock exchange prediction).
11. Fractals
 - a) fractal geometry in multifractal stochastic volatility models that work in the context of dynamic turbulence used for example for modeling market fluctuations. [7].Fractal analysis can roughly be thought of as a way to characterize and investigate the properties (e.g. selfsimilarity) of irregular sets.
 - b) fractal networks useful to express fractal populations like franchising networks.

Notation

<!...>	comment
<>	template recognized by the context
< >	container of phrase (markup)
≡>	link to something external (outside area)
def /def	start-end of def inition
spec /spec	start-end of specification
beg end	start-end of section

kkkk[..]	list of keywords assigned to kkkk
iiii(..)	list of items assigned to iiii
=	value assignment
@	mark of special attribute,feature,property
@dark	unknown, to obtain, to discover
::	belongs to
:	equivalent name (e.g.shortname)
#	number of
xxxx~<state>	e.g.<action>~(delayed,intime)
% mmm	macro
name	executive/operational object
ppppYyyy	name of group Yyyy with prefix 'pppp'
xxxx-nnnn	nnnn is an extension of xxxx
XXXX	basic object
UUUU.xxxx	structured name (xxxx object belonged to UUUU class)
& /	conjunctions 'and' 'or'
<-> ¹ or ↔	Bidirectional unitary relation 1:1
<- or ←	Back directional unitary relation 1:1
-> or →	Forward directional unitary relation 1:1
<=> <=>	Many to many relations

KERNEL

The kernel contains standard phrases and keywords common for subsets and environment. A special attention is paid to the object relations. Jay W.Forrester author of fundamental work "Industrial Dynamics"[4] underlined in [18] the importance of relations: "the structure of interconnections and the interactions are often far more important than the parts of system".

def OSL

```

def ENVIRONMENT: ENV
  ENV[regulations, infrastructure:INFR]
  regulations[legalacts,board-of-directors-resolutions,decisions]
  INFR[it,org,hr]:[itINFR,orgINFR,hrINFR]
  itINFR[servers,opersystems,applications,databases,users,prlanguages]
  orgINFR<!org. structure of subject>
/def
def globalMapping
  def subLang[BSL,HSL,SSL]<!subsets of OSL>
  objList<!list of objects>[area,subject,problem,decision,<def ined objects>]
  def List<!list of def initions>
  classList<!list of object classes>
  specList<!list of specifications i.e. descriptions of object instances>
/def

```

def (AREA e.g. BANKING, subject <NAME><!main object name e.g. >/
def <name><!ordinary object/item name,singular and plural>
 keywords:kwords<! keywords added during implementation >
 object.id<!object identifier>
 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[input(parameters,data),output(info,messages),
 structure(addComponent,addRelations)]
 incObject<!incarnation of object>,
 bindObject:Binder<!collection of integrated objects>,
 bookObject<!e.g.book of orders>
 copyObject<!copy of object>
 probObject:PROBLEM<!task to be performed>,
 interObject<object created by intersection of objects>,
 capsObject:CAPSULE<!portion of information reserved for a given user>]

<!dynamics def inition>

event:ev<!--elementary atomic fact >
 interval[dt<!decision time>,ts<!transit>]
 operation:op
 action:ac<!sequence of operations or events>
 process:pr<!sequence of actions and events including intervals for decision,transit etc.>
 pr[trigger,<actions><events>,endEvent]
 dynamics[event,operation/transaction,action,process]
 dynamics[ev,op/tr,ac,pr]<!short notation>
 dynamics.scenario[evSc,opSc,acSc,prSc]<!event scenario,...>
 trans<!transaction in terms of operating system monitor>
 ftrans<!financial transaction>
 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>

 olh<!object life history>[timeline,events,aging-curve]

<!interactions-relations>

role[interface,integrator,component,monitor,commander,
 driver,trigger,reactor,agent,executor,generator
 locator,executor/performer,initiator,terminator,destructor,

participator,owner,stockholder,customer,supplier;partner,employee]

relations[activated by,activates,assisted by,built from ,
appearance 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]

state[active,inactive,dark,dormant,suspended,aborted,delayed,intime,
variable,invariable,idle/waiting,lost,expected,deleted,finished,
rejected]

status[generic,real,virtual,undef ined]<!ontological status>

reactor[acceptance,rejection,constructor]

rank[critical,necessary,most wanted,optional,worst,best]

rule[decision-table,when-if,formula,heuristic].

/def

def gProfile<!geometric profile>

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

def spiral

spiral[single-strand,multi-strand]

spiral[helix,logarithmic,hyperbolic,polygonal,rational,golden,
spherical,conical,ellipse-involute,cornu,daisy,epispiral,
archimedian,fermat,nielsen,ulam,poinot,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)]

/def spiral

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>]

ellipse[surEllipse, edgeEllipse]

cylinder[edgeCylinder,insCylinder:tunnel]

edgeCylinder<!population on the edge of Cylinder>

insCylinder<!population inside of cylinder>

```

triangle<!layout 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]
/def gProfile
def neural-network
  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]2
/def neural-network

```

def control-flow

```

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[algebraic-iteration,spring,spiral]
activated by <.> with <initial-value> at <time-point>
  when <condition>
finished at <> with <...> when <.>
/def

```

```

/def kernel

```

BSL - Business Specification Language

BSL is focused on business objects. There are many businesses. To illustrate the power of language we take two of them. The widest spectrum of activities exists in manufacturing industry. In [4] are specified six main flows: material, energy, orders, finance, human resources and information. Information flow links all streams giving the overall picture of enterprise. Banking belongs to very complicated business if taking into account not only simple products like accounts, deposits and loans but also derivatives, forex, flow cash projection and risk management.

def BSL

```
objective[lterm<!longterm>,sterm<!shortterm]
: [ltermObjective,stermObjective]
decisions[critical,crucial]
curve[netIncome(year),netIncome(5year)sale(2year),inventory(1year),#customerClaim]
```

def BUSINESS[BANKING,INDUSTRY,TRADE, SERVICES]**def AREA** [BANKING]

```
def BANKING[RETAIL,WHOSALE,UNIVERSAL,
            MONEY-ARKET,DERIVATES,SHARES]
```

def ENV

```
bank.id(BIC<!Bank Identification Code>,
account.id[IBAN<!International Bank Account Number>,delivery-channels)
itSystem[application(<system.subsystems.modules>,
            itTools(math-modeling,bigdata,datamining,geometric-recognition)]
dataTables[Libor,OperatingCurreneces,ExchangeRates]
delivery-channels[internet-accounts(computer,smartphone,iTV),phoneline,branch,ATM]]
/def
```

```
keywords[customer.id,account,listAccount,accountCurrency,creationDate,cashFlow,
         fTrans<!financial-transaction>,fTransLimit]
```

```
problem[capital-assets-level,new-product-demand,customer-satisfaction]
```

<!geometric view>

```
spiral[multistrand( profit,competitiveness,credit-ability) on customer]
fractal(country,branch,ATM)
curve(capital-assets,netProfit,#customer,#branch,#ATM)
tunnel(x-loans,y-moy<!months-of-the-year>,z-credit ability)
```

def BANKING.retail

```
basicRetail.product[currACCOUNT,DEPOSIT,LOAN]
```

def subject BANK

```
object.Info[BIC,country,bCurrency<!base currency>,
            FinancialYear,#branches <!number of branches>]
dataTables[corrBanks,Branches,calendarWorkingDays,
            bkAccountChart,productList,interestRateTable]
keywords[branchNo,idCustomer,accountNo,rate,balance,balanceSheet]
OperationalObjects|teller,accountMgr,customerMgr,productMgr,trader|
```

```

Bank.objects[customer,product,currency,limit,account]
limit[countryLimit,industryLimit,customerLimit,currencyLimit]
cluster[profit-rate,growth-rate,competitiveness]<!Global factors for subject>
iObject[customerPosition,monthlyBalancesheet]<!for each customer>
typeBank[dmBank<!domestic>,frBank=>><!foreign>,corrBank=>><!correspondent>]
bkAccount[bsAccount<!balance-sheet>,nbsAccount<!nonbalance-sheetAccount>]
batchOperations[eodOperation<!at end of day>,eomOperation<!at end of month>,
    eoyOperation<!at end of year>,eopOperation<!at end of product>]
flow[cashFlow,#customer,#-customer-account,#private-banking-customer,#ATM,#branch]

```

```

def currACCOUNT<!current account>
    objectInfo[Account id,owner,co-owner,minBalance,actualBalance,historyStatement]
    Relates to idCustomer
    rtTrans[Open,Quit,Cash-in,Cash-out,transfer]<!real time transaction>
    eomOperation[monthlyStatement ]
/def

```

```

def CUSTOMER
    Belongs to <customer-segment> evaluated by dataMining/neural-network
    listAccount:(<list of accounts>),
    fTransLimit for each CUSTOMER,
    LOAN activated when accepted,
    customerPosition(#account,cashFlow,#overDrafts)
/def

```

```

/def subject>
def Banking.RETAIL

```

```

def AREA(INDUSTRY)

```

```

itSystem[application(<system.subsystems.module,program,build-block>),
    itTools(math-modeling,bigdata,datamining,geometric-recognition,forecasting)]
object[factory,product,distributor,retailer,consumer,order,resource,shipment,delivery]
resource(capital-equipment:cap-equip,material,energy,finance,labour)
problem[external-disturbances(new-technology,new-players),netIncome-decrease,
    inventory-reaction-to-sale,sale-forecast,market-demand]

```

```

order[rcvd<!received>,fct<!factory-inprogress>,forc<!forecast>,unfilled]:
    [rcvdOrder,fctOrder,forcOrder,unfilledOrder]
fractal[country,factory,sales-representative]

```

```

def flow <!inspirations taken from [4] >
    flow[business,finance,supplychain,material,energy,cap-equip,
        labour,production,task,control,info,data]
    : [businessFlow,financeFlow,supplychainFlow,materialFlow,energyFlow,
        labourFlow,productionFlow,taskFlow,controlFlow,infoFlow,dataFlow]
    volume/level(IN<increase>,OU<decrease>)
    rate(%formula)
/def

```



```
/def  
/def
```

```
def AREA(TRADE)<!to be def ined>  
  <!objects, trade flow,  
    financeFlow,infoFlow,labourFlow>
```

```
/def
```

```
def AREA(SERVICES)<!to be def ined>  
  <!objects, servicesFlow,  
    financeFlow,infoFlow,labourFlow>
```

```
/def
```

```
/def bsl>
```

Conclusion

BSL covers the business activity at general (business area, environment) and objects level in terms of structure, properties and processes. Usability of this approach would be increased by adding graphical presentation and mathematical models.

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,

books:

- [4] author = Forrester J.W.
title = Industrial Dynamics,
address = USA,publisher = The MIT Press, year = 1961,
- [5] author = Johnson R.A.,Cast F.E.,Rosenzweig J.E.
title = The Theory and Management of Systems.,
address = USA,publisher = McGraw-Hill Book Co, year = 1967,
- [6] author = Bowman K.P.
title = An Introduction to Programming with IDL: Interactive Data Language ,
address = USA, publisher = Elsevier Academic Press, year = 2006,
- [7] author = Mandelbrot, B.B.,
title = Fractals and Scaling in Finance:Discontinuity, Concentration, Risk,
address =New York, publisher = Springer-Verlag,: year = 1997

internet links:

- [8] www.w3.org/Markup/SGML/
- [9] www.cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Applications/daveapps.html
- [10] www.omg.org/spec/UML/2.5.1
- [11] [www.omg.org/spec/OCL/OCL Specification v2.4.1 \(2014\).](http://www.omg.org/spec/OCL/OCL%20Specification%20v2.4.1%20(2014))
- [12] www.yaml.org/spec/
- [13] webdesign.tutsplus.com/articles/an-introduction-to-structured-data-markup--webdesign-8577
- [14] www.latexeditor.org
- [15] www.sgmlsource.com/history/roots.htm
- [16] www.harrisgeospatial.com/SoftwareTechnology/IDL.aspx#graphics
- [17] [www.oasis-open.org/committees/human markup/documents/HM.Primary-Base-Spec-1.0.html](http://www.oasis-open.org/committees/human%20markup/documents/HM.Primary-Base-Spec-1.0.html)
- [18] pubsonline.informs.org/doi/pdf/10.1287/mnsc.14.9.601
- [19] www.hegel.net/en/e0.htm