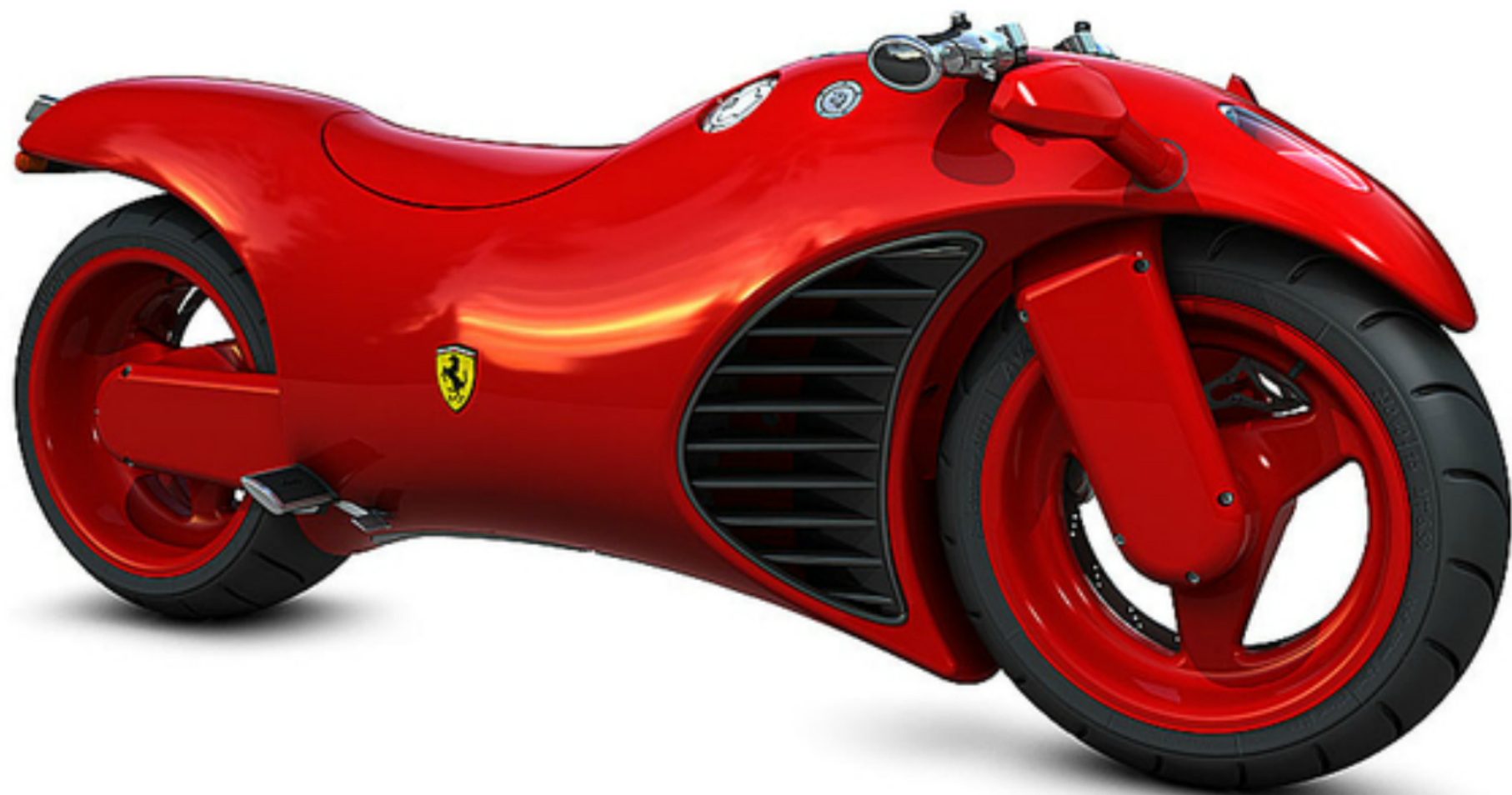


Domain Driven Design



Jonas.Andersson
@metria.se
@umejug
@agilaume





Another example is the following flight simulator, the winner of the 1998 IOCCC,^[20] as listed and described in *Calculated Bets: Computers, Gambling, and Mathematical Modeling to Win (1991)*^[21] and shown below:

```

#include <math.h>
#include <sys/time.h>
#include <X11/Xlib.h>
#include <X11/keysym.h>
double L , o , P
, _=dt,T,Z,D=1,d,
s[999],E,h= 8,I,
J,K,w[999],M,m,O
,n[999],j=33e-3,i=
1E3,r,t, u,v ,W,S=
74.5,l=221,X=7.26,
a,B,A=32.2,c, F,H;
int N,q, C, y,p,U;
Window z; char f[52]
; GC k; main(){ Display*e=
XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(O)){ struct timeval G={ 0,dt*1e6}
; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+= *P; r=E*K; W=cos( O); m=K*W; H=K*T; O+=D*_*F/ K+d/K*E*_; B=
sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T*E+ D*B*W; j+=d*_*D-_*F*E; P=W*E*B-T*D; for (o+=(I=D*W+E
*T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
] == 0|K <fabs(W=T*r-I*E +D*P) |fabs(D=t *D+Z *T-a *E)>K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2/ K
*D; N-1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_* (X*t +P*M+m*1); T=X*X+ 1*1+M *M;
XDrawString(e,z,k ,20,380,f,17); D=v/l*15; i+=(B *1-M*r -X*Z)*_; for(; XPending(e); u *=CS!=N){
XEvent z; XNextEvent(e ,&z);
+++((N=XLookupKeysym
(&z.xkey,0))-IT?
N-LT? UP-N?& E:&
J:& u: &h); --*(
DN -N? N-DT ?N==
RT?&u: & W:&h:&J
); } m=15*F/l;
c+=(I=M/ 1,l*H
+I*M+a*X)*_; H
=A*r+v*X-F*1+(
E=.1+X*4.9/l,t
=T*m/32-I*T/24
)/S; K=F*M+(
h* 1e4/l-(T+
E*5*T*E)/3e2
)/S-X*d-B*A;
a=2.63 /l*d;
X+=( d*1-T/S
*(.19*E +a
*.64+J/1e3
)-M* v +A*
Z)*_; l +=
K *_; W=d;
sprintf(f,
"%5d %3d"
"%7d",p =1
/1.7,(C=9E3+
O*57.3)%0550,(int)i); d+=T*(.45-14/l*
X-a*130-J* .14)*_/125e2+F* *v; P=(T*(47
*I-m* 52+E*94 *D-t*.38+u*.21*E) /1e2+W*
179*v)/2312; select(p=0,0,0,0,&G); v-=(
W*F-T*(.63*m-I*.086+m*E*19-D*25-.11*u
)/107e2)*_; D=cos(o); E=sin(o); } }

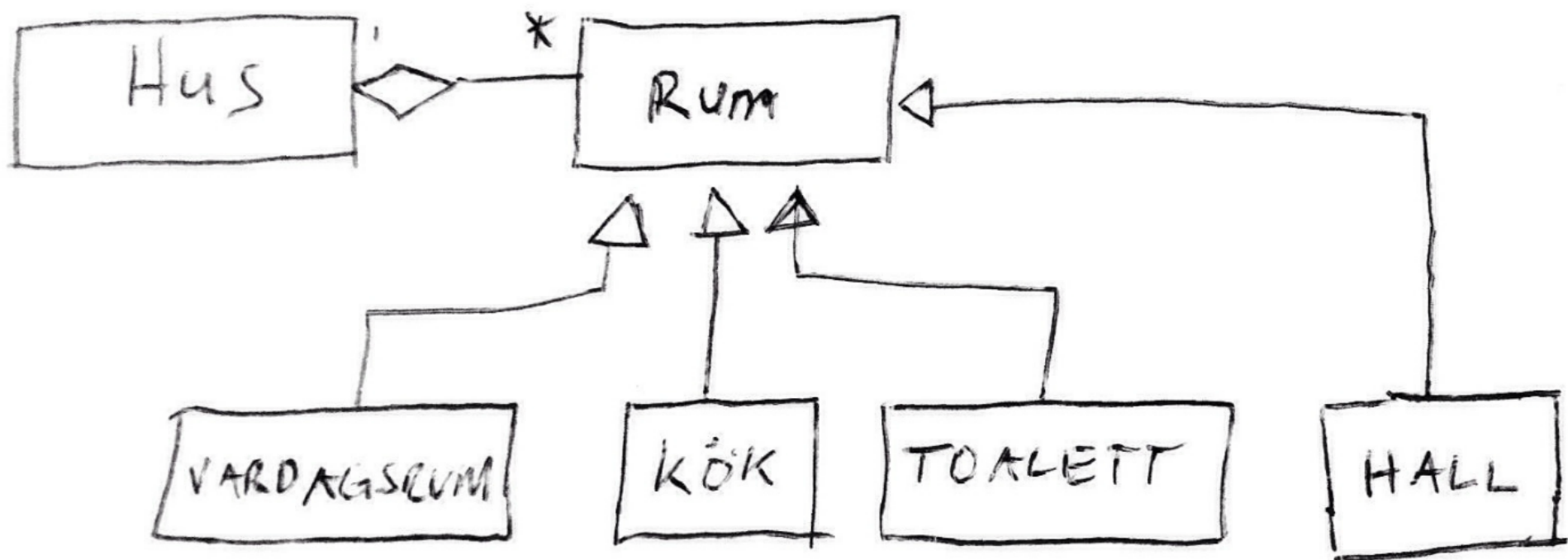
```

Vad är en Domän?

kunskapsområde

expertområde

sphere of knowledge



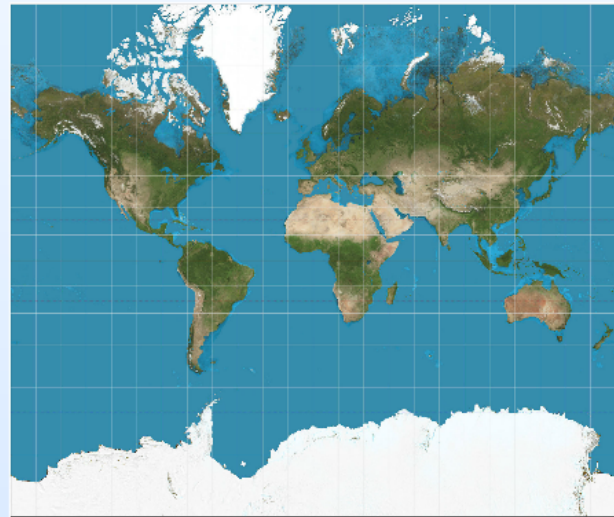
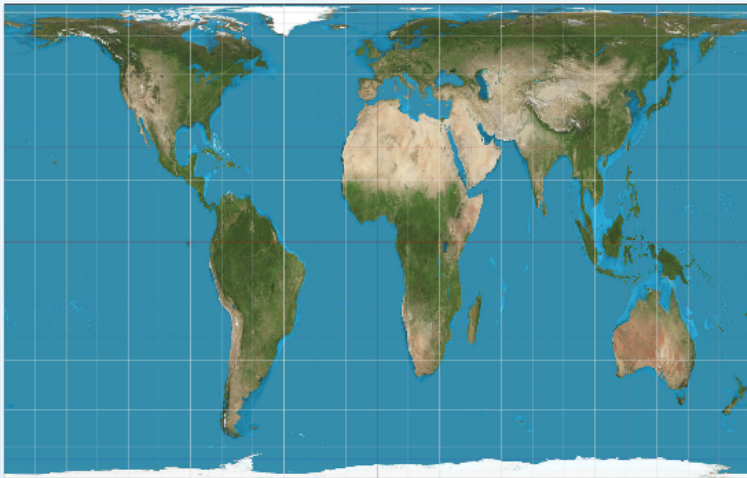
```
1 package se.metria.spikes.ddd;
2
3 import static org.fest.assertions.Assertions.assertThat;
4
5
6
7
8
9
10 public class HusTest {
11
12     public class Hus {
13         private List<Rum> rum = new ArrayList<Rum>();
14
15         public List<Rum> rum() {
16             return rum;
17         }
18
19         public void add(Rum rum) {
20             this.rum.add(rum);
21         }
22     }
23
24     public class Rum {
25     }
26
27     public class Vardagsrum extends Rum {
28     }
29
30     public class Kök extends Rum {
31     }
32
33     public class Toalett extends Rum {
34     }
35
36     @Test
37     public void shouldContainRum() throws Exception {
38         Hus hus = new Hus();
39
40         Rum vardagsrum = new Vardagsrum();
41         Rum kök = new Kök();
42         Rum toalett = new Toalett();
43
44         hus.add(vardagsrum);
45         hus.add(kök);
46         hus.add(toalett);
47
48         assertThat(hus.rum().size()).isEqualTo(3);
49         assertThat(hus.rum()).contains(vardagsrum);
50     }
51 }
```

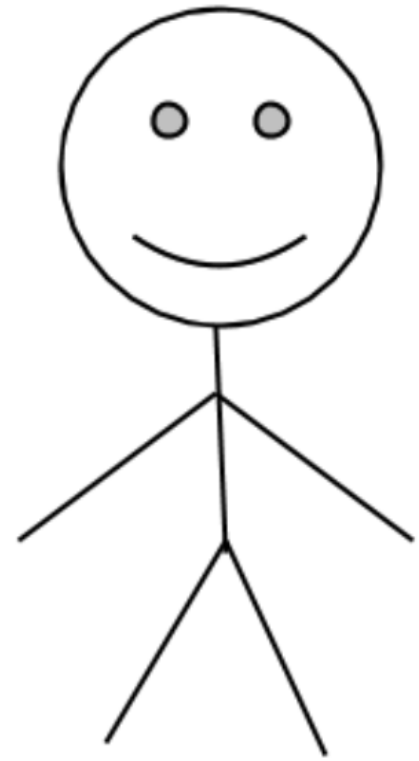
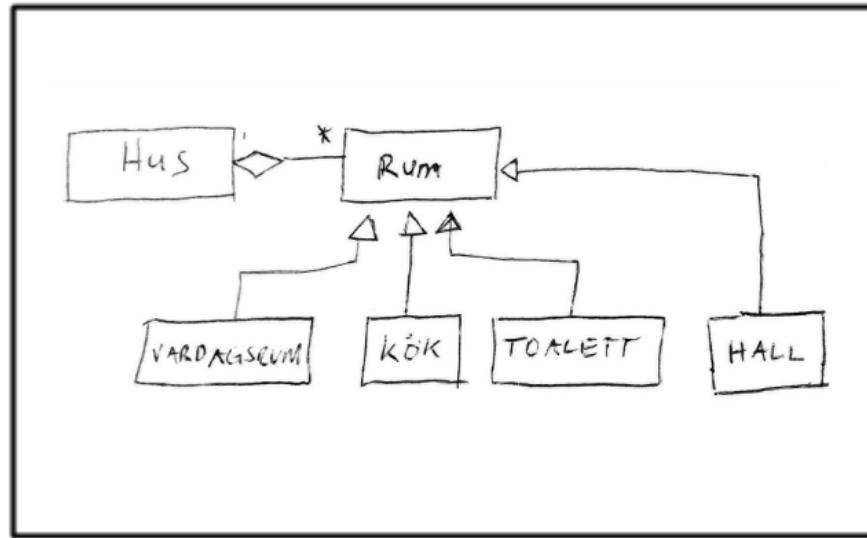
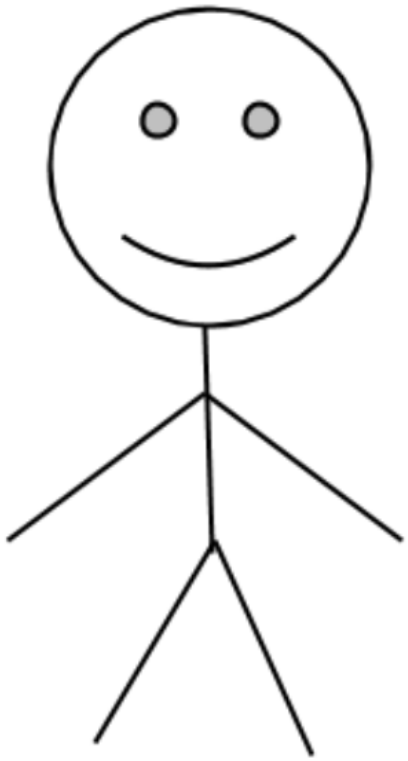




Stateless

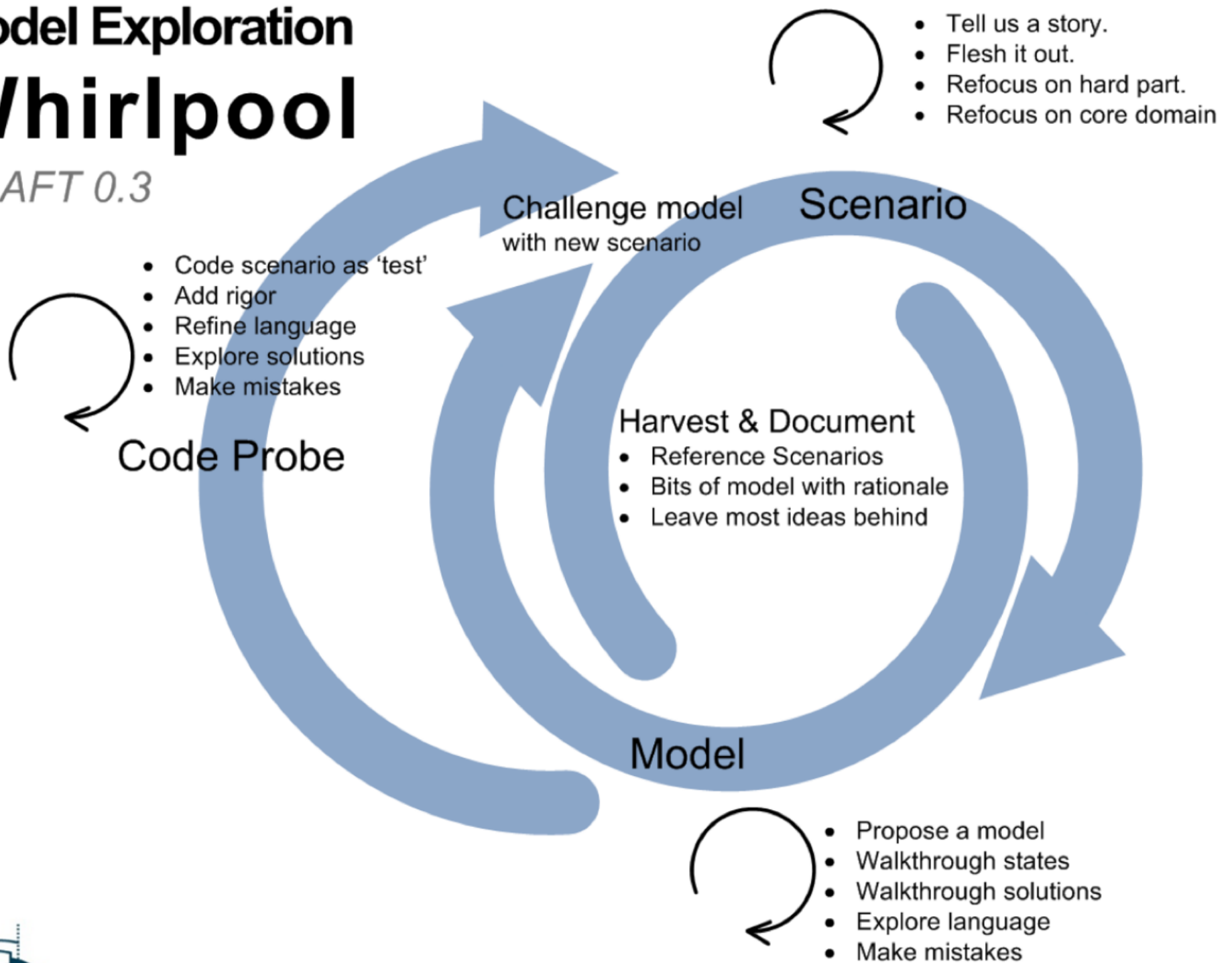
Kartor





Model Exploration Whirlpool

DRAFT 0.3



Domain Driven Design

Ubiquitous Language

Domain Model

Supple Design

Core Domain

Vad är DDD?

Principer för mjukvaruutveckling:

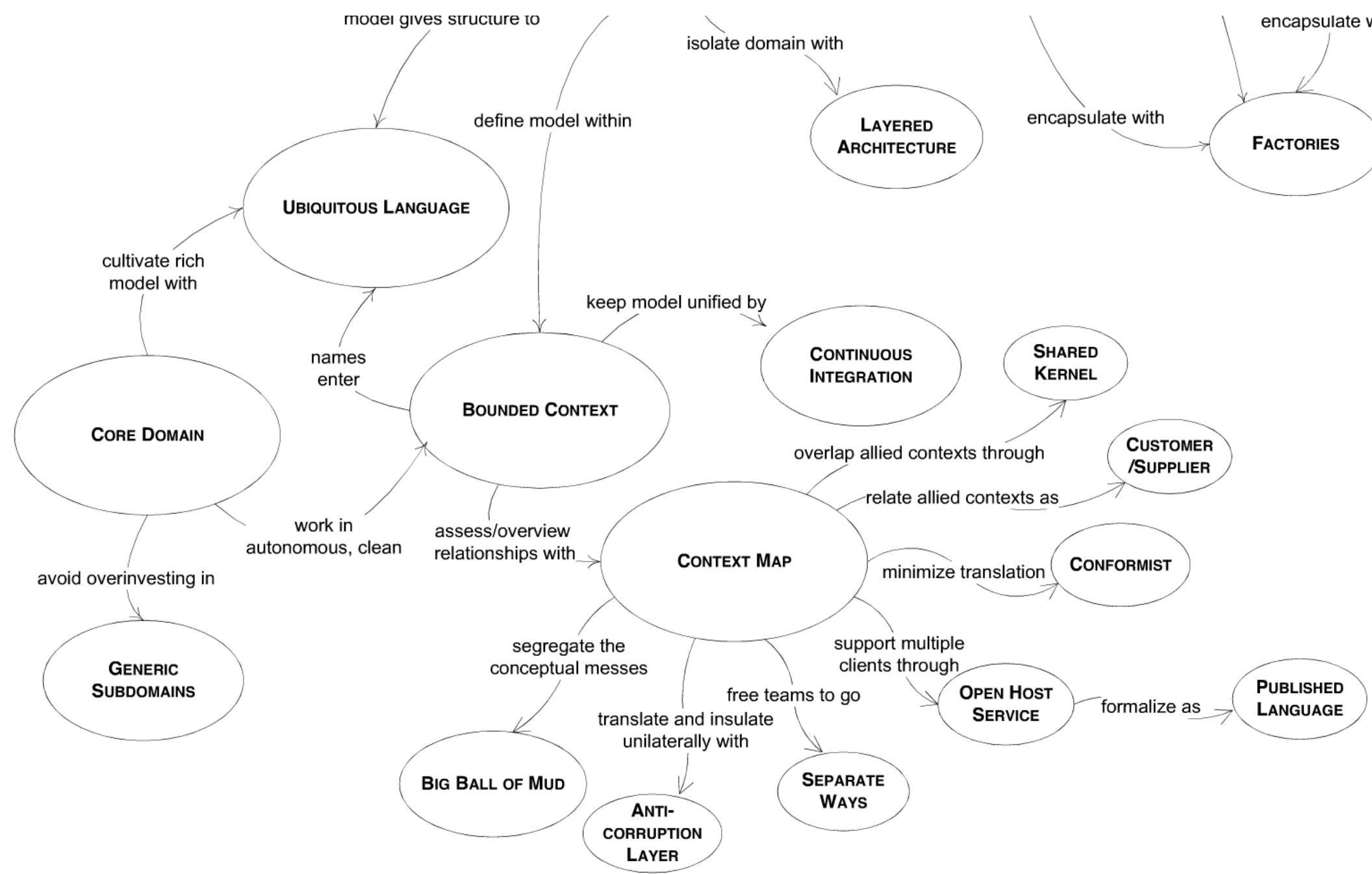
- Prata samma språk - ubiquitous language
- Etablera en domänmodell
- Utforska modellen tillsammans
- Fokusera på core domain

Domain-Driven Design Reference

Definitions and Pattern Summaries



by Eric Evans

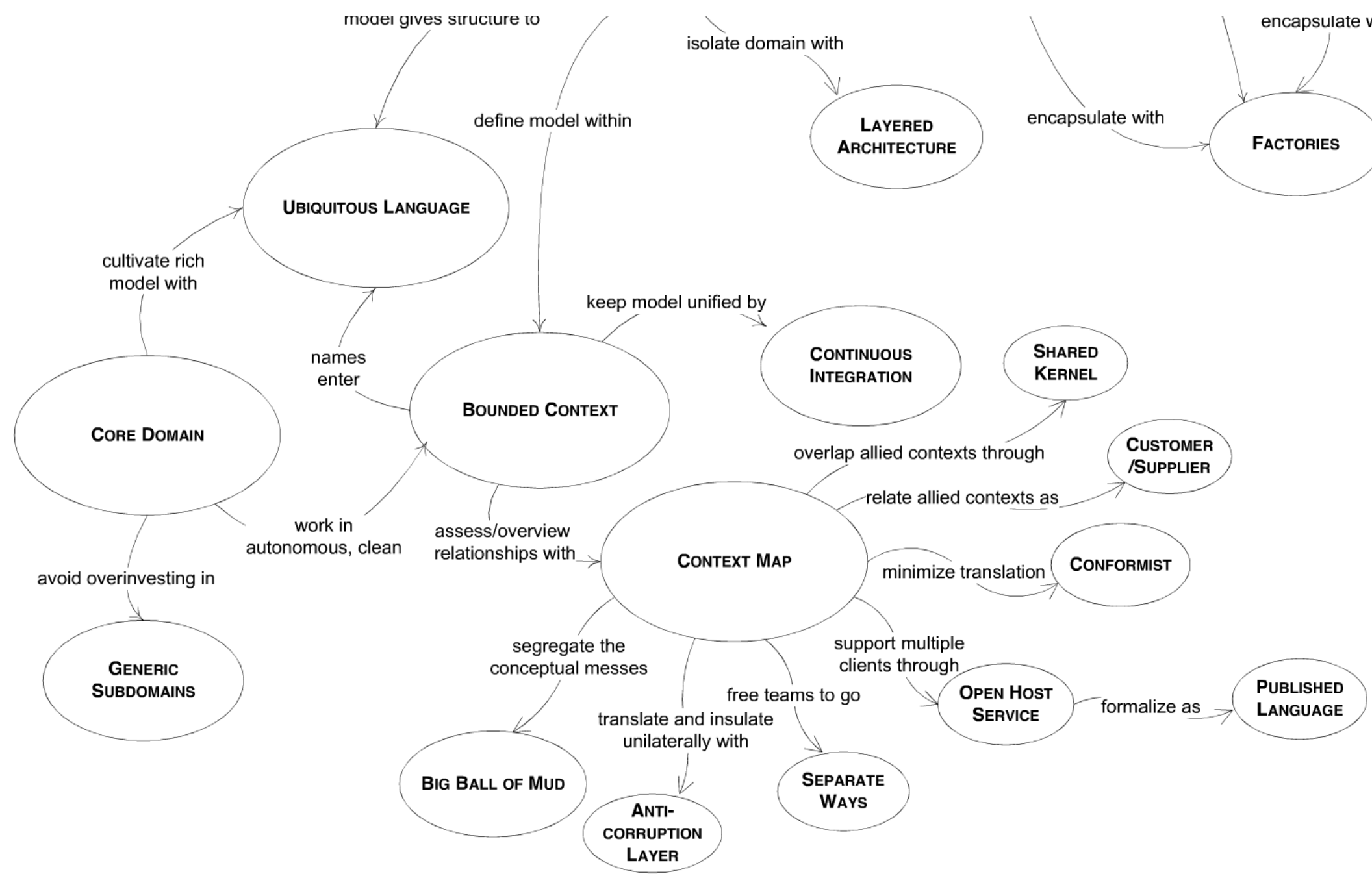


by Eric



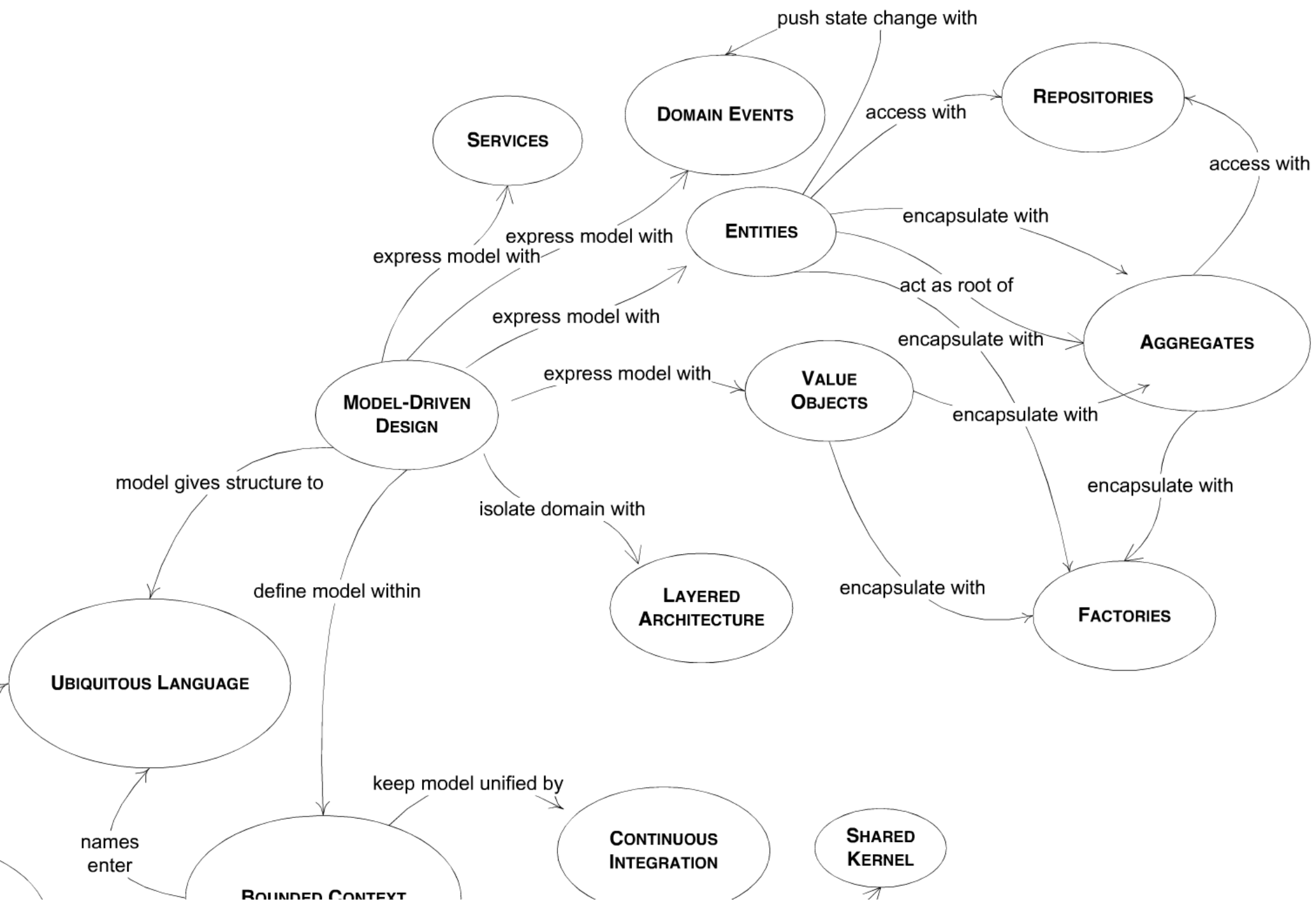
Bord för fvra



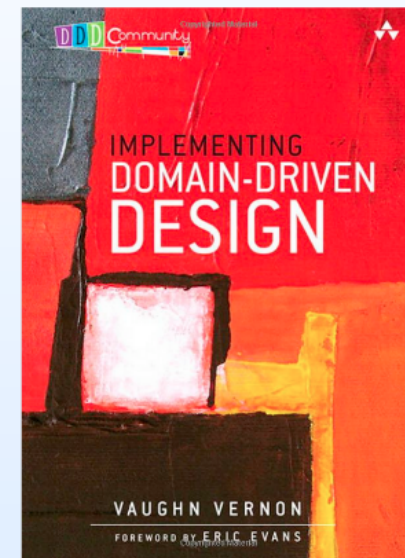
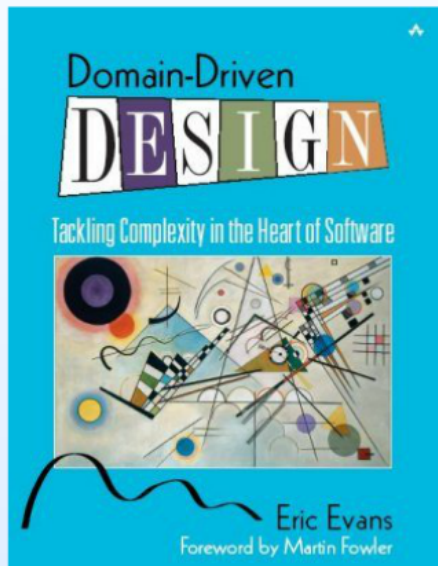


by Eric

and Pattern Summaries



Mer om DDD



dddcommunity.org

jonas.andersson@metria.se