FERI
Univerza v Mariboru
Fakulteta za elektrotehniko,
računalništvo in informatiko

DOMENSKO SPECIFIČNI MODELIRNI JEZIKI 2024/2025

2. predavanje

Ponovitev prejšnjega tedna

- Primeri DSLjev
- Anatomija modernih urejevalnikov kode
- Razdelitev programskih jezikov
- Definicije

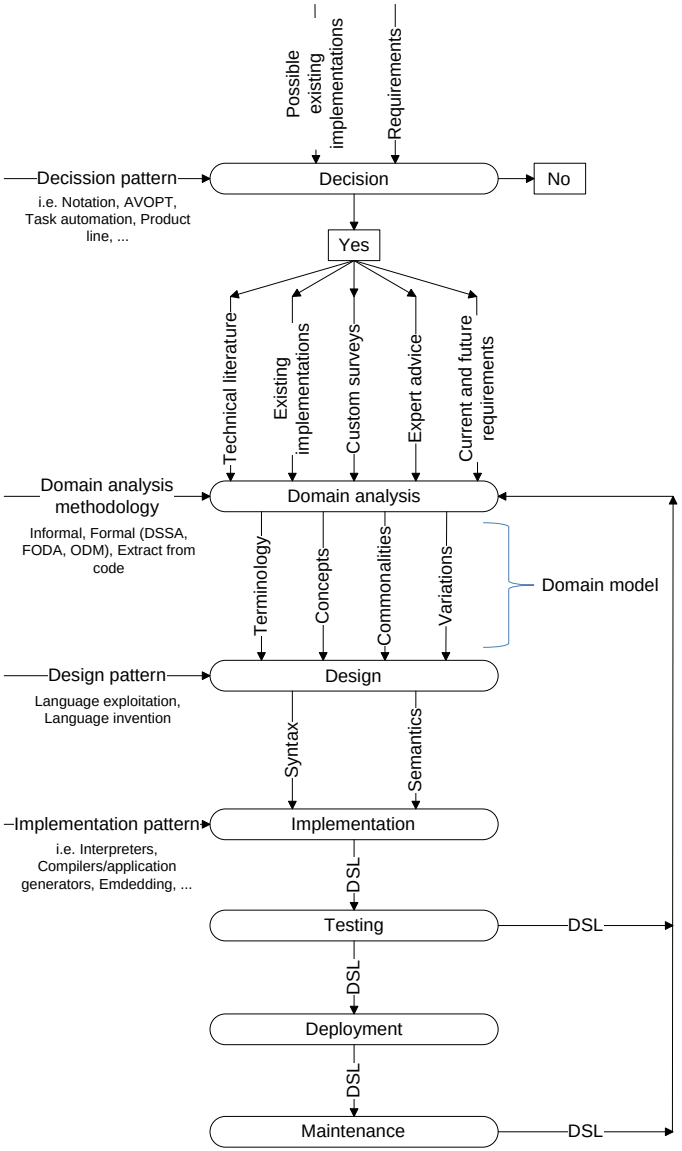
Seminarske vaje – izbira domene

- Danes, 4. 10. 2024, ob 14.00, DELTA

- Izbira domene
  - Ko je domena odobrena, lahko pričnete z implementacijo naloge 1.2
  - Spletna aplikacija
  - Uporaba sklada MERN
  - Več (glej prosojnice vaj)

Življenjski cikel razvoja DSLjev

- Marjan Mernik, Jan Heering, and Anthony M. Sloane. 2005. When and how to develop domain-specific languages. *ACM Comput. Surv.* 37, 4 (December 2005), 316-344.

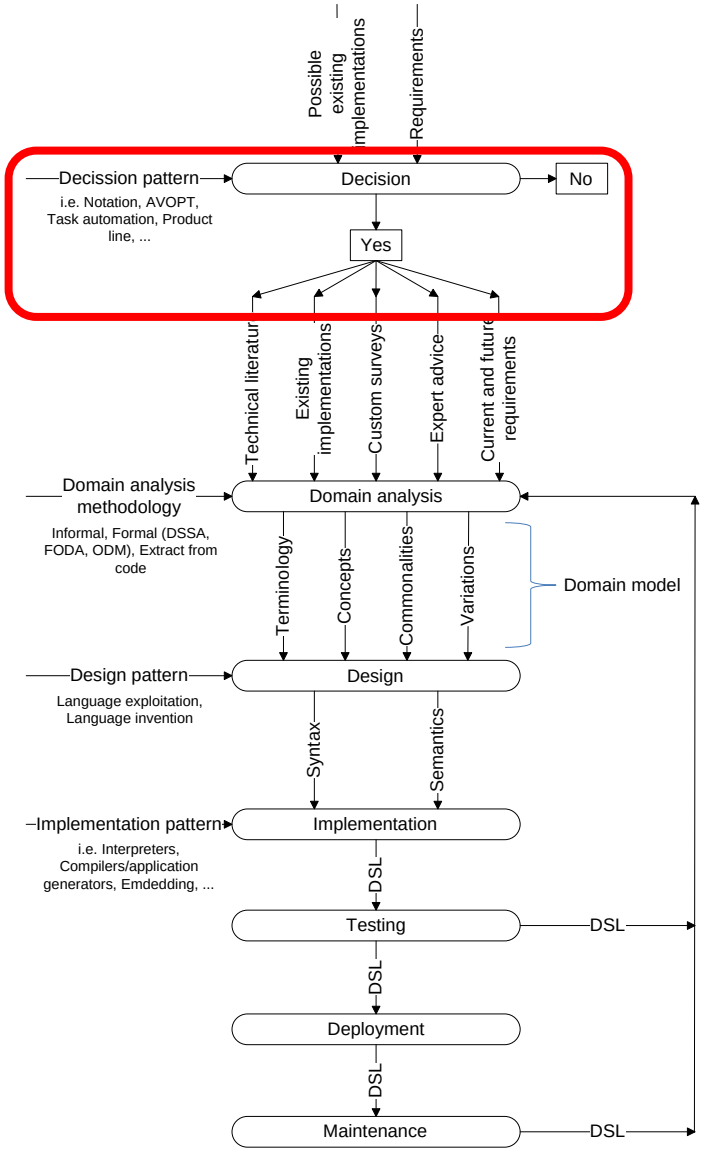


Faza odločitve razvoja DSLjev

- Razvoj novega DSL-ja je lahko drag:
  - Razvoj DSL je težek, saj zahteva strokovno **znanje o domeni in razvoju jezika**. Malo razvijalcev ima oboje.
  - **Razvojne tehnike** DSLjev so bolj **raznolike** od tistih za splošno-namenske jezike (GPLje) in zahtevajo skrbno preučitev vseh vključenih dejavnikov.
  - Rast števila uporabnikov, usposabljanje, spremembe, podpora razvoja, itd. – **vzdrževanje postane** resna in časovno zahtevna **težava**.

Življenjski cikel razvoja DSLjev

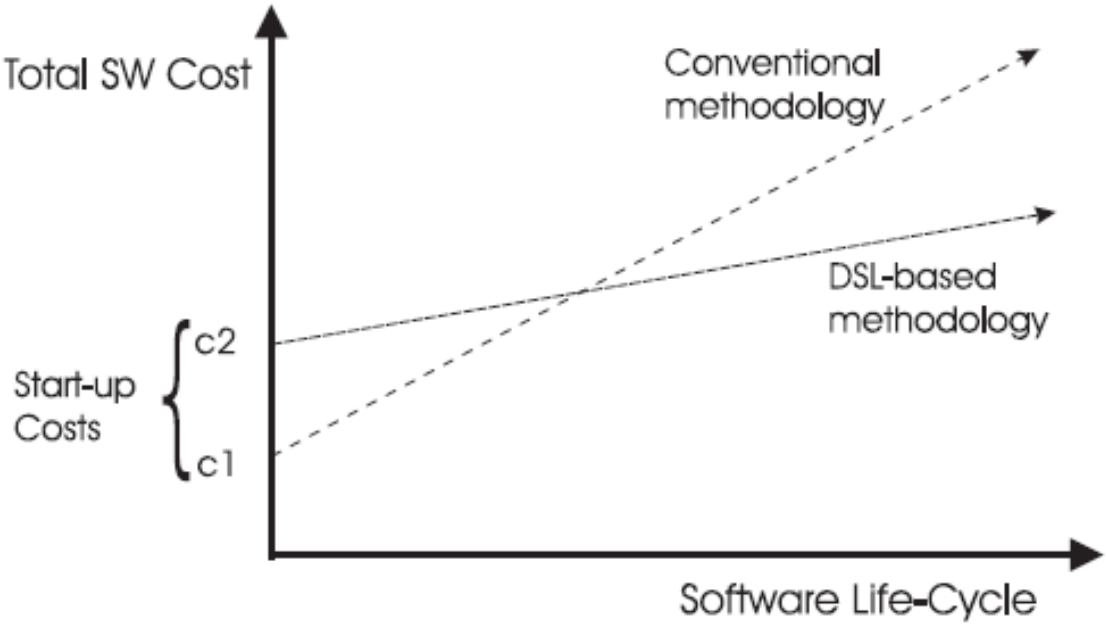
- Marjan Mernik, Jan Heering, and Anthony M. Sloane. 2005. When and how to develop domain-specific languages. *ACM Comput. Surv.* 37, 4 (December 2005), 316-344.



DSL – razvoj da ali ne?

- To niso edini dejavniki, ki otežujejo odločitev za razvoj novega DSLja. Na začetku pogosto ni jasno, **ali bo DSL koristen** ali da bo razvoj novega DSLja **stroškovno upravičen**.
- To lahko postane jasno šele po **izvedeni naložbi v rešitev GPL** in njeno uporabo. Ko je jasno, da programer operira z domensko specifičnimi konstrukti, lahko razmislimo o **alternativni rešitvi** v obliki DSLja.
- V takih primerih je razvoj DSLja lahko ključni korak pri **prenovi programske opreme** ali **razvoju programske opreme**.

Začetni stroški razvoja pri DSL



Kdaj se izplača investicija v DSL

Vir slike: P. Hudak. Modular Domain-Specific Languages and Tools. ICSR'98

Vzorci odločitev za razvoj DSLjev

- **Vzorci odločitve identificirajo primere**, kjer je razvoj novega DSLja smiselen.
- Osnova vseh odločitvenih vzorcev je podobna – pomisleki, kot so:
  - **zmanjšan strošek razvoja** programske opreme,
  - **omogočen razvoj** programske opreme, ki obsega veliko domenskega in manj strokovnega (programerskega) znanja ali celo **razvoj programske opreme končnih uporabnikov** z znanjem iz določeno domeno, vendar praktično brez strokovnega (programerskega) znanja.

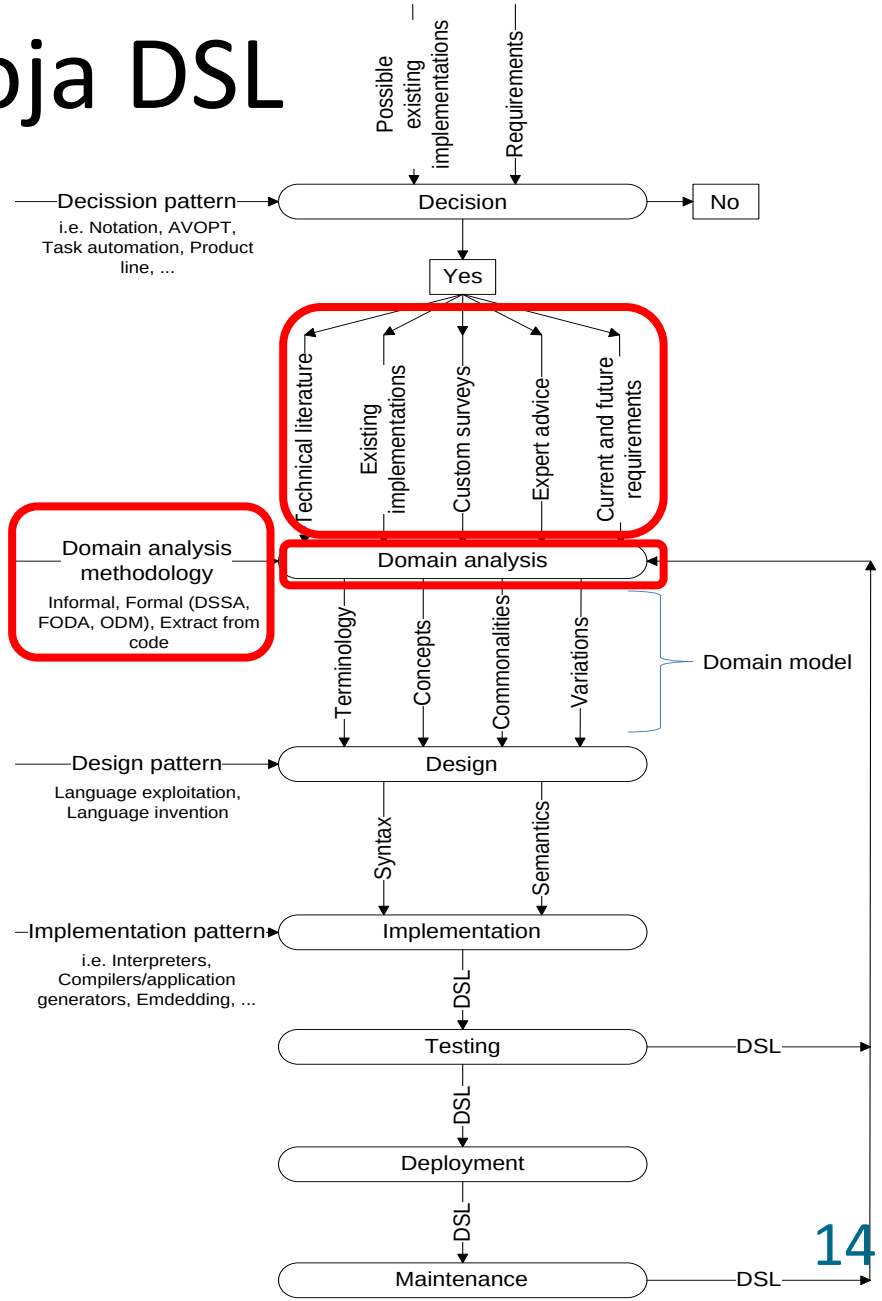
Vzorci odločitev za razvoj DSLjev

- Vzorci odločitev
  - **Notacija** (odločilni dejavnik je razpoložljivost ustreznih, obstoječih ali novih notacij specifičnih za posamezno področje);
    - primeri: dodati uporabniku prijazno notacijo za obstoječi API, preoblikovanje vizualne v tekstovno notacijo
  - **Avtomatizacija nalog** (opisovanje nenavadnih in ponavljajočih se programskih nalog)
    - Npr. pogojni stavki
  - **Predstavitev strukture podatkov** (opis zapletene podatkovne strukture)
    - Npr. XML, JSON
  - **Sprehodi po strukturi podatkov** (opisovanje prehodov po podatkovni strukturi)
    - Npr. XPath

Vzorci odločitev za razvoj DSLjev

- Vzorci odločitev
  - **Čelni del sistema** (opis sistema konfiguracij)
  - **Interakcija** (opis interakcijo s sistemom)
  - **Razvoj GUI** (pogosto opravljena z DSLjem)
  - **Vzorec AVOPT** (opisuje analizo domene, verifikacijo, optimizacijo, paralelizacijo in transformacijo)
  - **Produktna linija** (DSL uporabimo za zapis specifikacij produktne linije)

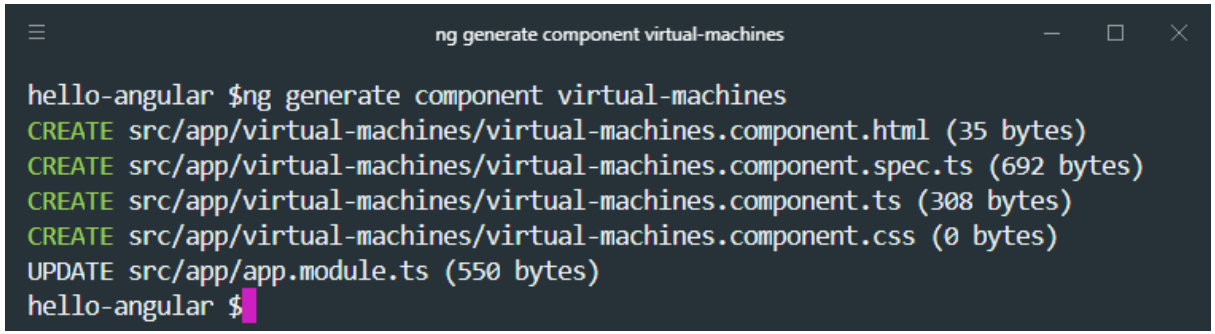
Življenjski cikel razvoja DSL



Analiza domene

- DSL je **računalniški jezik**, namenjen določeni domeni.
- **Zagotavlja** ustrezne **vgrajene abstrakcije in notacije**.
  - Potrebno je narediti **analizo domene**.

- **Programi v DSL** eksplicitno določajo **le del obnašanja** – velik del obnašanja je impliciten in določen.
  - Ugotoviti moramo **fiksni in variabilni del domene**.



Kaj je domena?

- Domena: znanje ali aktivnosti na področju za katerega je značilen **niz konceptov** in **terminologija**, ki ju razumejo strokovnjaki na tem področju.

- V **programskem inženirstvu** (angl. software engineering) domeno pogosto razumejo kot **družino sistemov**, tj. niz programskih sistemov, ki imajo podobno funkcionalnost.

Znanje o domeni - ontologija

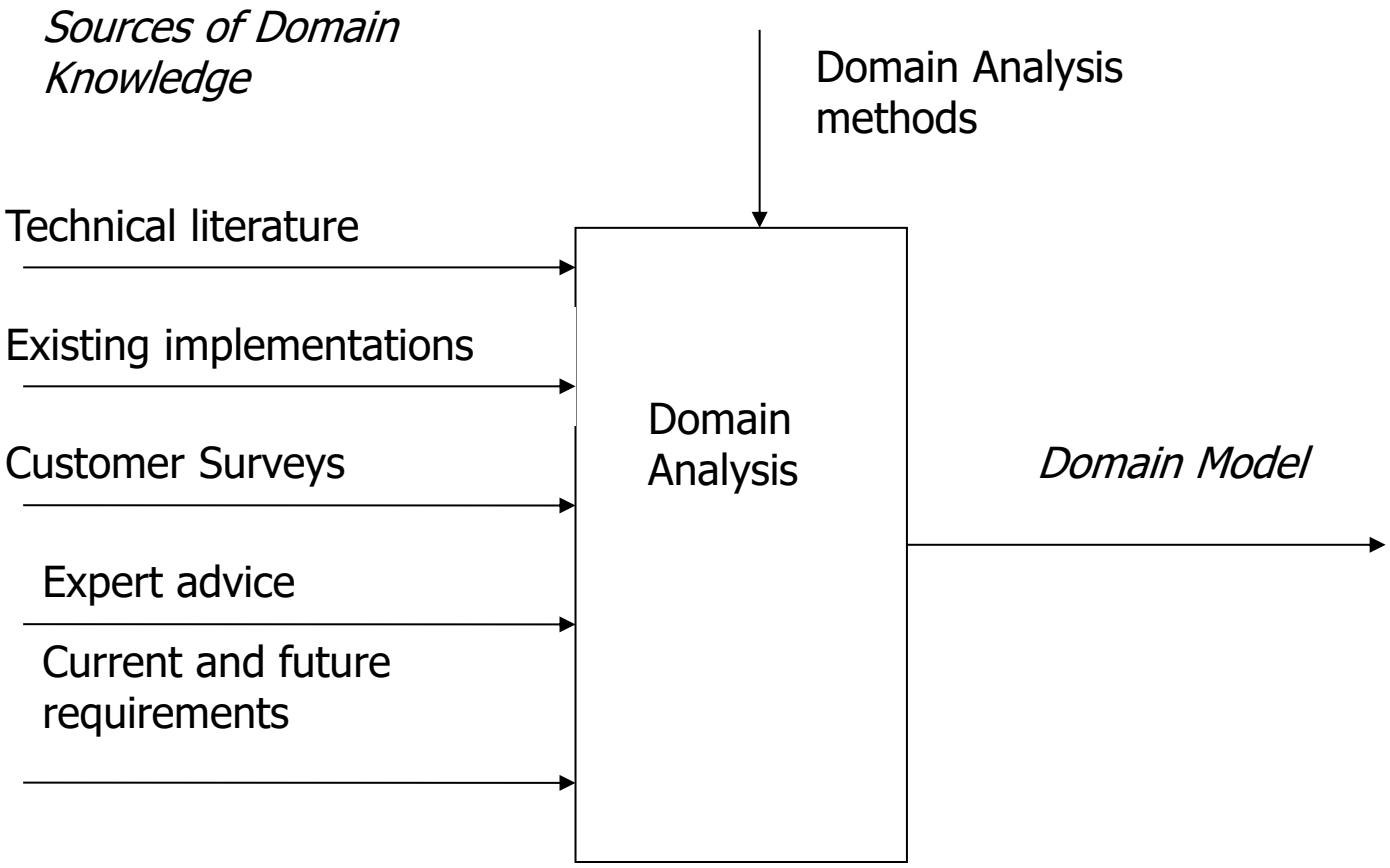
- Ontologija je **ogrodje znanja**, ki opisuje določeno domeno z uporabo **reprezentativnega besednjaka**.
- Ontologija določene domene vsebuje:
  - **terminologijo**,
  - **koncepte**,
  - **taksonomijo**,
  - **njihove odnose** in
  - **domenske aksiome**.

Napačna prepričanja

- Kdaj imamo opravka z domeno?
  - **Nekaj** (aplikacijskih) **primerov uporabe** ni dovolj za specializacijo in klasifikacijo domene.

- Je **vsak skriptni jezik** tudi domensko specifični?
  - Skriptni jeziki so pa lahko splošno-namenski kot tudi domensko-specifični.
  - Podobno, modelirni in paralelni jeziki so lahko tako splošno-namenski kot domensko-specifični.
- **Skriptiranje**, na primer, je način **uporabe jezika** in ne **domena aplikacije**.

Vhodi v domensko analizo



Vhodi v domensko analizo

- Proces identifikacije, analize in predstavitve
- **Domenski model** in **arhitekturo** DSLja dobimo iz študije:
  - obstoječih tehničnih poročil,
  - obstoječih sistemov (implementacij),
  - vprašalnikov,
  - znanja ekspertov,
  - nastajajoče in obstoječe zahteve (zgodovine razvoja v določeni domeni).

- **Namen** domenske analize je:
  - izbira in določanje **obsega domene**,
  - zgraditi **model domene**.

Domenska analiza

- Obstajajo mnoge metodologije domenske analize:
  - **FODA – Feature-Oriented Domain Analysis (razvita na Software Engineering Institute)**
  - ODM – Organization Domain Modeling (M. Simos)
  - Draco (J. Neighbors)
  - DARE – Domain Analysis and Reuse Environment (W. Frakes & R. Prieto-Diaz)
  - DSSA – Domain-Specific Software Architecture (ARPA)
  - FAST – Family-Oriented Abstraction, Specification, and Translation (D. Weiss)
  - ODE – Ontology-based Domain Engineering (Falbo et al.)

FODA – modeli lastnosti

- Temelji na **modelih lastnosti** (angl. feature models)
- Modeli lastnosti se uporabljajo v domenski analizi **za zajem skupnih in variabilnih delov v domeni**.

- Modeli lastnosti so sestavljeni iz:
  - **Diagram lastnosti (angl. feature diagram)**: predstavlja **hierarhično razgradnjo lastnosti** in njihovih vrst (obvezne, neobvezne in opcijske lastnosti)
  - **Definicije lastnosti**: opisujejo vse lastnosti (semantika)
  - **Kompozicijska pravila za lastnosti**: opišimo katere kombinacije so veljavne/neveljavne
  - **Utemeljitev lastnosti**: razlogi za izbiro lastnosti.

Diagram lastnosti (angl. feature diagram)

- Diagram lastnost je predstavitev sistema končnega uporabnika

- Diagram lastnosti se sestoji iz:
  - **Lastnosti**
  - **Kompozicijskih pravil**

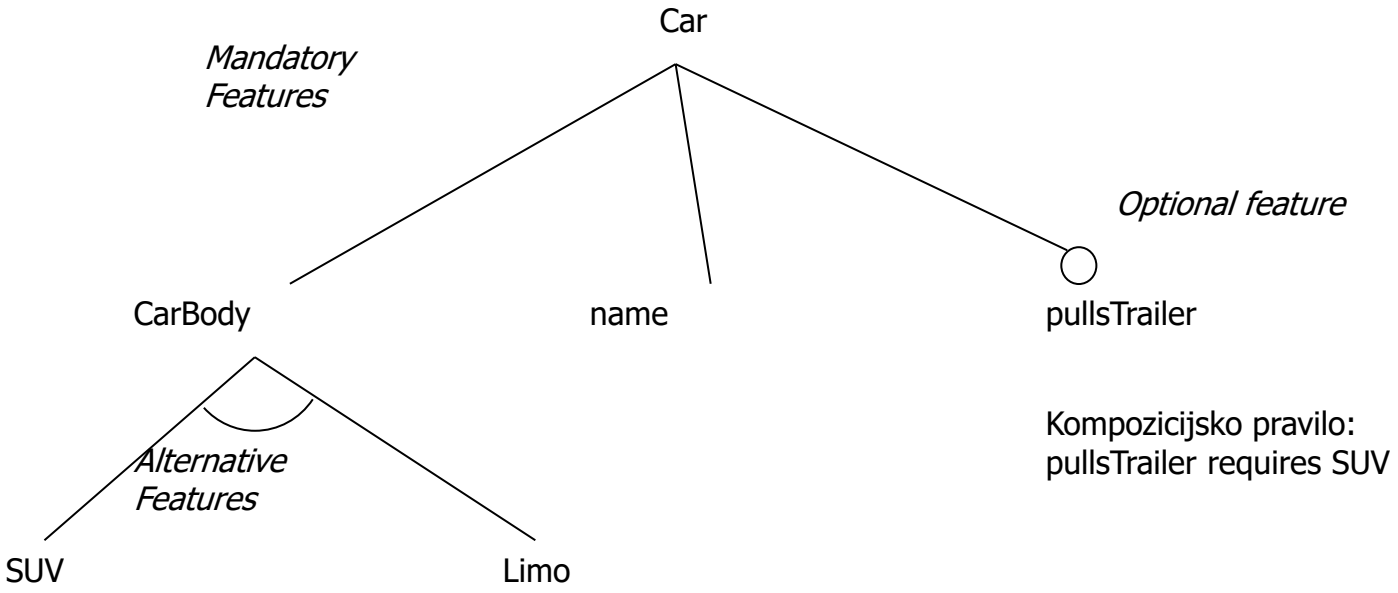
Diagram lastnosti

- **Lastnosti** (angl. features):
  - **Obvezne** (angl. mandatory features): sistem v domeni mora vsebovati to lastnost.
  - **Neobvezne** (angl. optional fetures): sistem lahko vsebuje lastnost ali pa ne.
  - **Alternative** (angl. alternative feature): sistem lahko vsebuje samo eno lastnost naenkrat.
  - **Ali** (angl. or-feature): sistem lahko vsebuje več kot eno lastnost.

- Soodvisnosti med lastnostmi so predstavljene s **kompozicijskimi pravili** (angl. composition rules):
  - **Zahteva** (angl. require): zajema posledice med lastnostmi
  - Vzajemno **izključevanje** (angl. mutually-exclusive rules): omejitve v modelu glede kombinacij funkcij

Diagram lastnosti

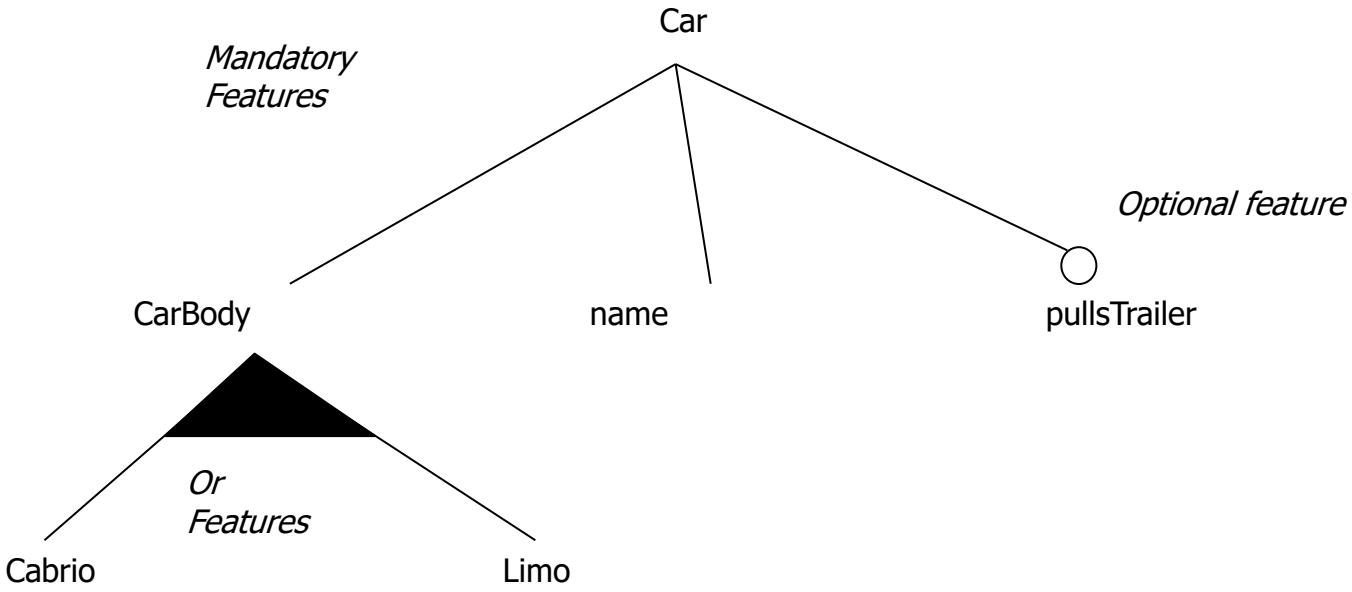
- **Prvi primer** diagrama lastnosti



Rationale:
Limo is more fuel efficient

Diagram lastnosti

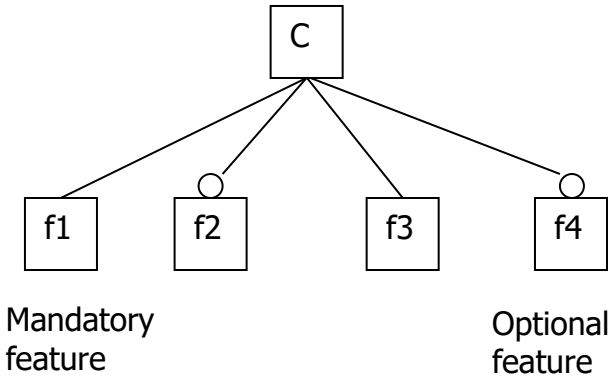
- **Prvi primer** diagrama lastnosti



Cabrio + Limo =
convertible

Diagram lastnosti

- **Korensko vozlišče** (C) predstavlja **koncept**, ostale vozlišča pa predstavljajo lastnosti
- **Obvezna** lastnost (npr. f1) je vključena v koncept, če so vključene tudi njene starševske lastnosti
- **Neobvezna** lastnost (f2) je lahko vključena v koncept, če in samo če je vključena tudi njena starševska lastnost.

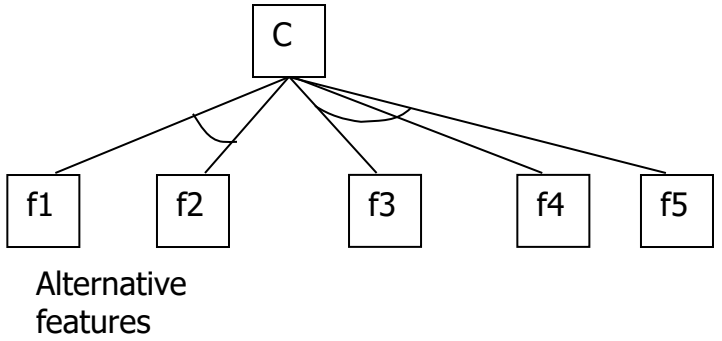


{f1, f2, f3}
{f1, f2, f3, f4}
{f1, f3}
{f1, f3, f4}

Primeri programov

Diagram lastnosti

- **Alternativne** lastnosti
  - angl. one-of
  - alternative
  - v koncept je vključena **natančno ena lastnost** iz niza alternativnih lastnosti.

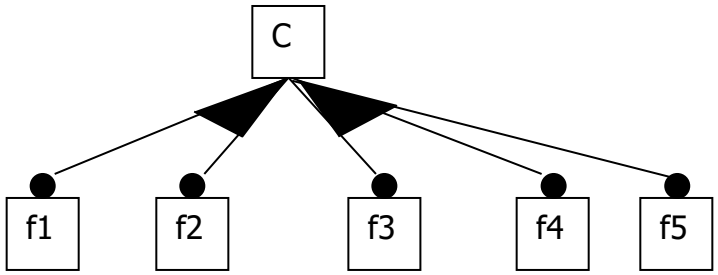


{f1, f3}
{f1, f4}
{f1, f5}
{f2, f3}
{f2, f4}
{f2, f5}

Primeri programov

Diagram lastnosti

- **Ali** lastnosti
  - angl. or-features, more-of
  - vključena je **vsaka neprazna podmnožica** iz niza lastnosti.



Or- features

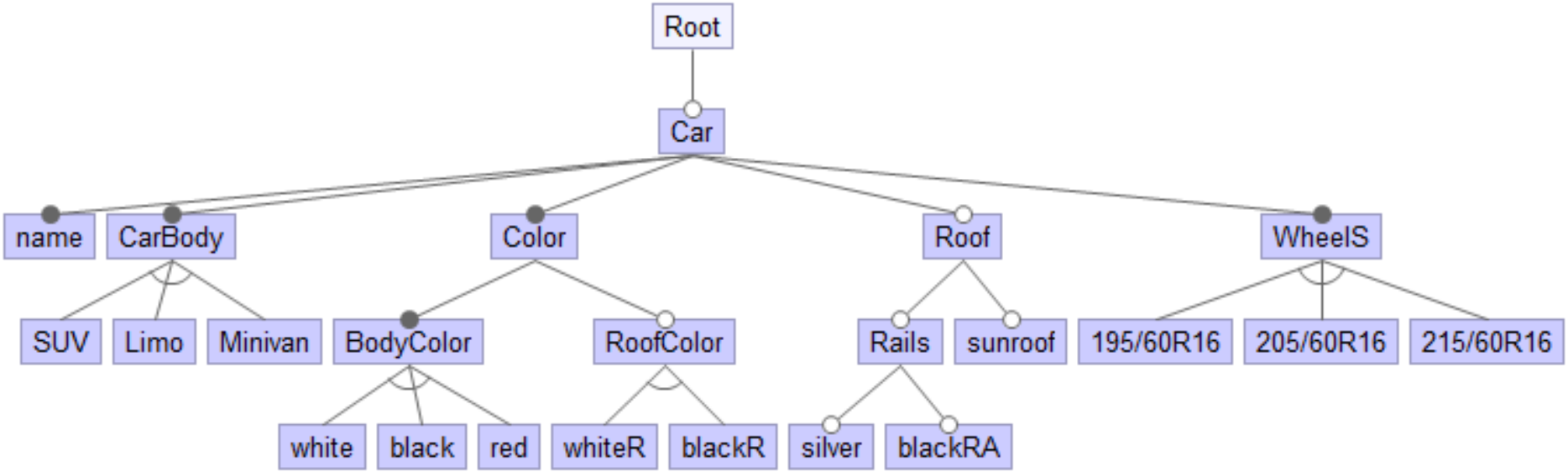
{f1, f3}
{f1, f2, f3}
{f1, f3, f4}
{f1, f3, f5}
{f1, f2, f4, f5}
{f1, f2, f3, f4, f5}
...

Primeri programov

Diagram lastnosti

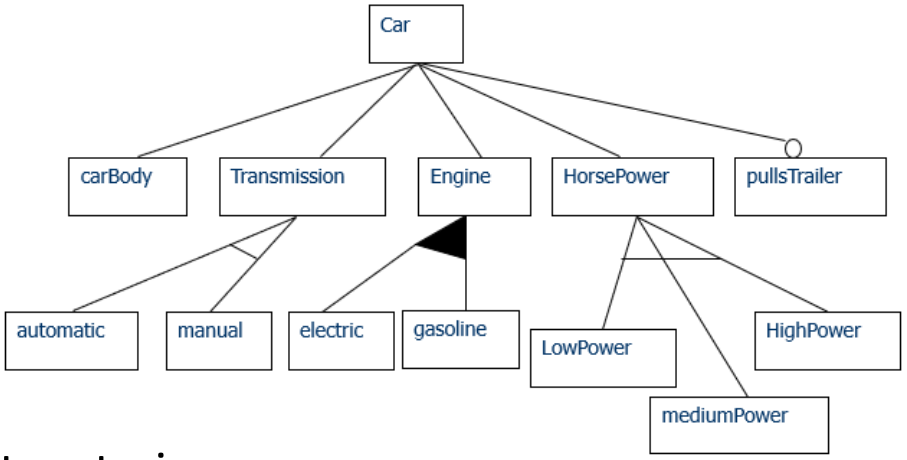
```
car "MyCar2"
    carbody Limo
    color white
    roof black
    roof rails silver
```

- Primer:
  - Diagram lastnosti in program



  - Kaj manjka programu?

Diagram lastnosti



- Delitev glede na **pozicijo**:
  - **Atomarne lastnosti** ni mogoče dodatno razčleniti z drugimi lastnostmi.
  - **Sestavljene lastnosti** (angl. composite features) so lastnosti, ki so določene z drugimi lastnostmi.
- Delitev glede na **vključenost**:
  - **Skupna lastnost** koncepta je lastnost, ki je **prisotna v vseh primerkih koncepta**. Vse obvezne lastnosti, katerih **starš je koncept**, so skupne lastnosti. Tudi vse obvezne lastnosti, katerih **starši so skupne lastnosti**, so skupne lastnosti.
  - **Variabilnost** v diagramu lastnosti je izražena z **neobveznimi, alternativnimi in ali lastnostmi**. Te lastnosti imenujemo tudi **variabilne lastnosti**. Vozlišča, na katere so te lastnosti pripete, se imenujejo **variabilne točke**.

Diagram lastnosti

- Diagrami lastnosti se uporabljajo za analizo domene, da zajamemo:
  - **Skupne** (lastnosti) **značilnosti**
  - **Variabilne značilnosti** (opcijske, alternativne in ali lastnosti)

- Diagrami lastnosti natančno opisujejo **vse možne konfiguracije sistema** (imenovane tudi **primerki domene**), s poudarkom na **variabilnih lastnostih (**razlike konfiguracij).

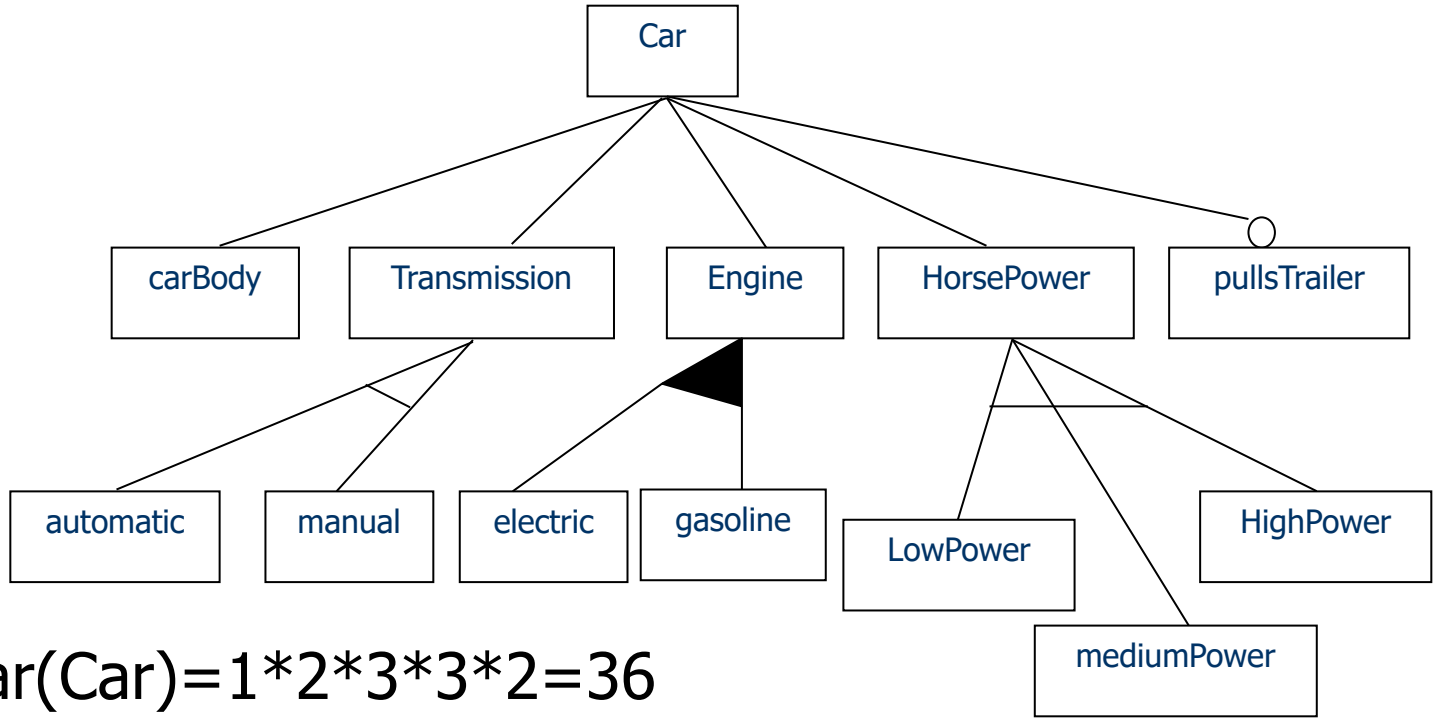
Diagram lastnosti

- Pravila variabilnosti - štejemo število možnosti za določen diagram lastnosti

Var(A)=1 // atomic feature
Var(F?)=var(F)+1 // optional feature
Var(one-of(F1 … Fn)) = var(F1) + …+ var(Fn) //alternative feature
Var(more-of(F1 … Fn)) = (var(F1)+1) * …* (var(Fn)+1) -1 //or feature
Var(C is F1 … Fn) = var(F1)* …* var(Fn) // concept

C-koncept
A-atomarna lastnost
F-sestavljene lastnosti

Diagram lastnosti



Var(Car)=1*2*3*3*2=36

```
Var(A)=1
Var(F?)=var(F)+1
Var(one-of(F1 … Fn)) = var(F1) + …+ var(Fn) feature
Var(more-of(F1 … Fn)) = (var(F1)+1) * …* (var(Fn)+1) -1
Var(C is F1 … Fn) = var(F1)* …* var(Fn)
```

Diagram lastnosti

- Vse možnosti različnih programov

```
(carBody, automatic, electric, lowPower, pullsTrailer),
(carBody, automatic, electric, gasoline,lowPower,
 pullsTrailer),
(carBody, automatic, gasoline, lowPower, pullsTrailer),
(carBody, automatic, electric, mediumPower,
 pullsTrailer),
(carBody, automatic, electric, gasoline,mediumPower,
 pullsTrailer),
(carBody, automatic, gasoline, mediumPower, pullsTrailer),
(carBody, automatic, electric, highPower, pullsTrailer),
(carBody, automatic, electric, gasoline,highPower,
 pullsTrailer),
(carBody, automatic, gasoline, highPower, pullsTrailer),
(carBody, manual, electric, lowPower, pullsTrailer),
(carBody, manual, electric, gasoline,lowPower,
 pullsTrailer),
(carBody, manual, gasoline, lowPower, pullsTrailer),
(carBody, manual, electric, mediumPower, pullsTrailer),
(carBody, manual, electric, gasoline,mediumPower,
 pullsTrailer),
(carBody, manual, gasoline, mediumPower, pullsTrailer),
(carBody, manual, electric, highPower, pullsTrailer),
(carBody, manual, electric, gasoline,highPower,
 pullsTrailer),
(carBody, manual, gasoline, highPower, pullsTrailer),
(carBody, automatic, electric, lowPower),
(carBody, automatic, electric, gasoline,lowPower),
(carBody, automatic, gasoline, lowPower),
(carBody, automatic, electric, mediumPower),
(carBody, automatic, electric, gasoline,mediumPower),
(carBody, automatic, gasoline, mediumPower),
(carBody, automatic, electric, highPower),
(carBody, automatic, electric, gasoline,highPower),
(carBody, automatic, gasoline, highPower),
(carBody, manual, electric, lowPower),
(carBody, manual, electric, gasoline,lowPower),
(carBody, manual, gasoline, lowPower),
(carBody, manual, electric, mediumPower),
(carBody, manual, electric, gasoline,mediumPower),
(carBody, manual, gasoline, mediumPower),
(carBody, manual, electric, highPower),
(carBody, manual, electric, gasoline,highPower),
(carBody, manual, gasoline, highPower))
```

Diagram lastnosti
3D vizualizacija klaviatur

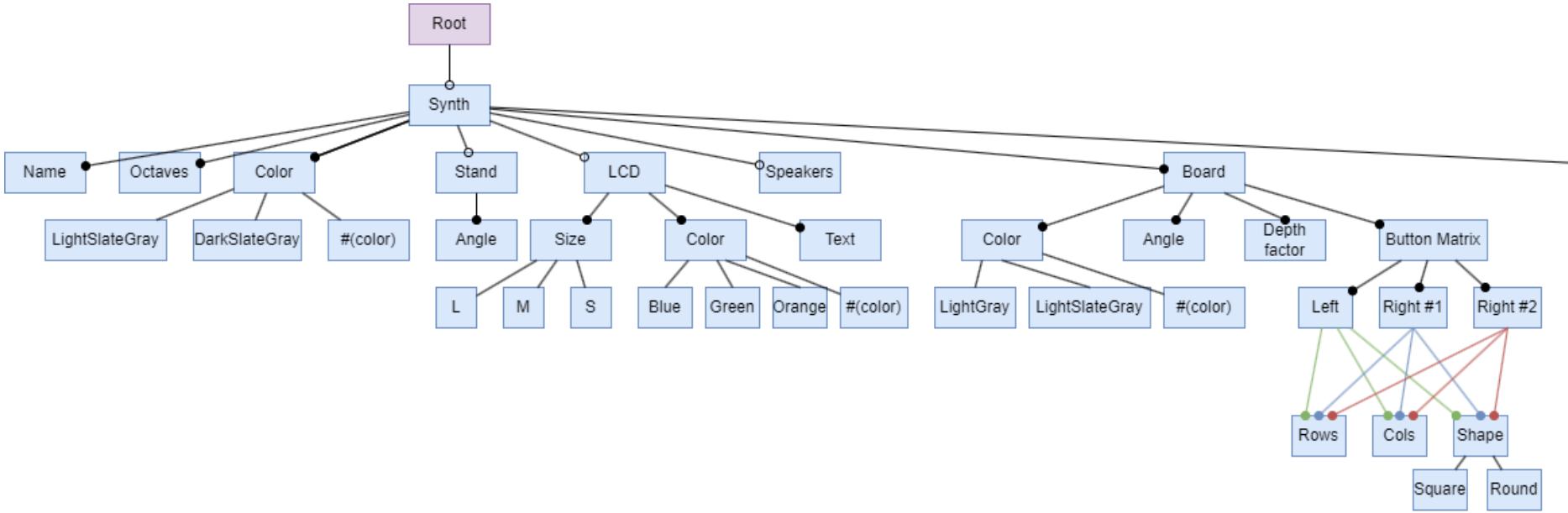
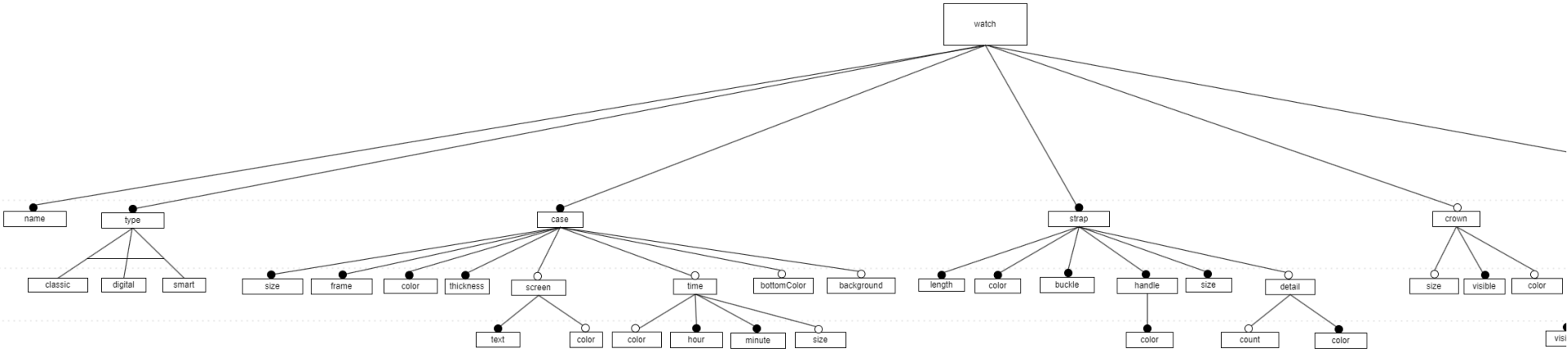


Diagram lastnosti
3D vizualizacija ročne ure



Domenska analiza

- Domena in diagrami lastnosti
  - Kaj vidimo iz diagrama lastnosti?
    - Skupne, spremenljive lastnosti, koncepte.

- Model domene je sestavljen iz:
  - Eksplicitne predstavitve **skupnih in spremenljivih lastnosti** sistema v domeni,
  - **domenskih konceptov,**
  - **semantike**,
  - **odvisnosti** med lastnostmi.

Domenska analiza

- **Domenski model** sestavljajo:
  - **Definicija domene**: določa obseg domene in karakterizira njeno vsebino s **podanimi primeri**, **napačnimi primeri** in **generičnimi pravili** za vključitev ali izključitev.
  - **Terminologija**: definira domenski **besednjak**.
  - **Model konceptov**: **opis** konceptov domene v **določenem formalizmu ali neformalnem** besedilu.
  - **Modeli lastnosti**: opis **skupnih in spremenljivih lastnosti** konceptov in njihove **soodvisnosti**. Modeli lastnosti predstavljajo konfiguracijski vidik konceptnih modelov.

Domenska analiza

- Nekatere **tipične aktivnosti** v domenski analizi:
  - Analiza **podobnosti**: analiziramo podobnosti med lastnostmi, entitetami, aktivnostmi, dogodki, relacijami, strukturami, ipd.
  - Analiza **variacij**: analiziramo razlike med lastnostmi, entitetami, aktivnostmi, dogodki, relacijami, strukturami, ipd.
  - Analiza **kombinacij**: analiziramo kombinacije lastnosti, ki nakazujejo tipične strukturne ali vedenjske vzorce.

Domenska analiza

- Priporočilo, da se analiza domene opravi **formalno** (npr. FODA).
- **Običajno** se med razvojem DSLja analiza domene opravi **neformalno**.
- Kako se lahko DSL razvije iz podatkov, zbranih **iz analize domene**?

Koncepti v domenski analizi

- Upoštevajte, da med analizo domene identificirani koncepti niso vedno uporabni za reševanje dejanskega problema.
- Zato se identificirani koncepti lahko klasificirajo v:
  - **Nepomembni –** niso pomembni za problem;
  - **Spremenljivi –** jih je dejansko treba opisati v programu DSL;
  - **Skupni –** jih je mogoče vgraditi v DSL.

Za vaje – naloga 2.1.

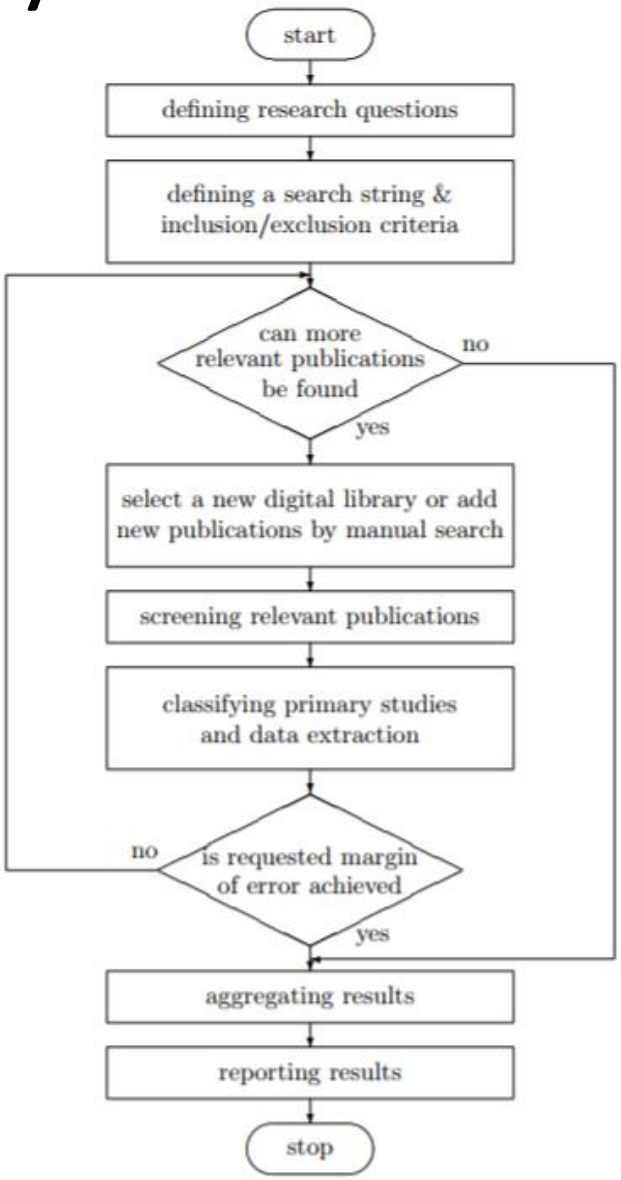
- V diagramski tehniki diagram lastnosti (angl. feature diagram) predstavite jezik iz naloge 1.1 in 1.2.

- FODA - Feature diagrams
- Orodje:
  - Evelance
  - Feature diagram tools
  - Draw

Za vaje – primer DSL EasyWorkFlow

- Latex – okolje „picture“

```
\begin{picture}(12.000000,23.000000)(-4.750000,-23.000000)
\put(1.2500,-0.5000){\oval(2.5000,1.0000)}
\put(0.0000,-1.0000){\makebox(2.5000,1.0000)[c]{\shortstack[c]{
start
}}}
\put(1.2500,-1.0000){\vector(0,-1){0.5000}}
\put(-2.7500,-2.5000){\framebox(8.0000,1.0000)[c]{\shortstack[c]{
defining research questions
}}}
\put(1.2500,-2.5000){\vector(0,-1){0.5000}}
\put(-2.7500,-5.0000){\framebox(8.0000,2.0000)[c]{\shortstack[c]{
defining a search string \&\\inclusion/exclusion criteria
}}}
\put(1.2500,-5.0000){\vector(0,-1){0.5000}}
\put(-1.7500,-7.0000){\line(2,1){3.0000}}
\put(-1.7500,-7.0000){\line(2,-1){3.0000}}
\put(4.2500,-7.0000){\line(-2,-1){3.0000}}
\put(4.2500,-7.0000){\line(-2,1){3.0000}}
\put(-1.7500,-8.5000){\makebox(6.0000,3.0000)[c]{\shortstack[c]{
can more\\relevant publications \\be found
}}}
...
```

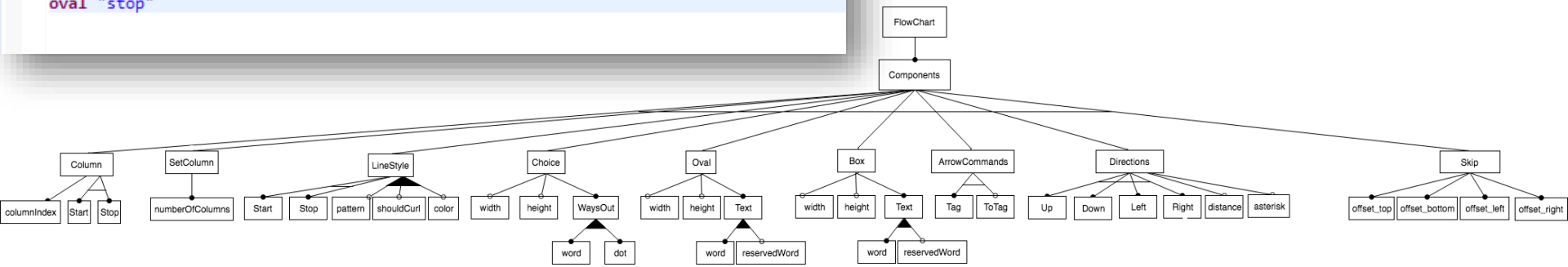


Za vaje – primer DSL EasyWorkFlow

- Iz napak se učimo - popravimo naslednji Feature diagram!

test.mydsl

```
default Skip 1 1 1 1
default Size 2 5

oval "start"
box "defining research questions"
box 4 5 "defining a search string & inclusion/exclusion criteria"

choice . no yes . "can more relevant publications be found"

box 4 5 "select a new digital library or add new publications by manual search"
box "screening relevant publications"
box 4 5 "classifying primary studies and data extraction"

choice . . yes no "is requested margin of error achieved"

box "aggregating results"
box "reporting results "
oval "stop"
```



Vprašanja