

Modeļvadāmas programmatūras izstrādes pamatidejas un terminoloģija

1. lekcija

Ērika Asņina, “Modeļvadāmas programmatūras izstrādes
praktikums”, RTU, 2011

Lekcijas saturs

■ Modeļi

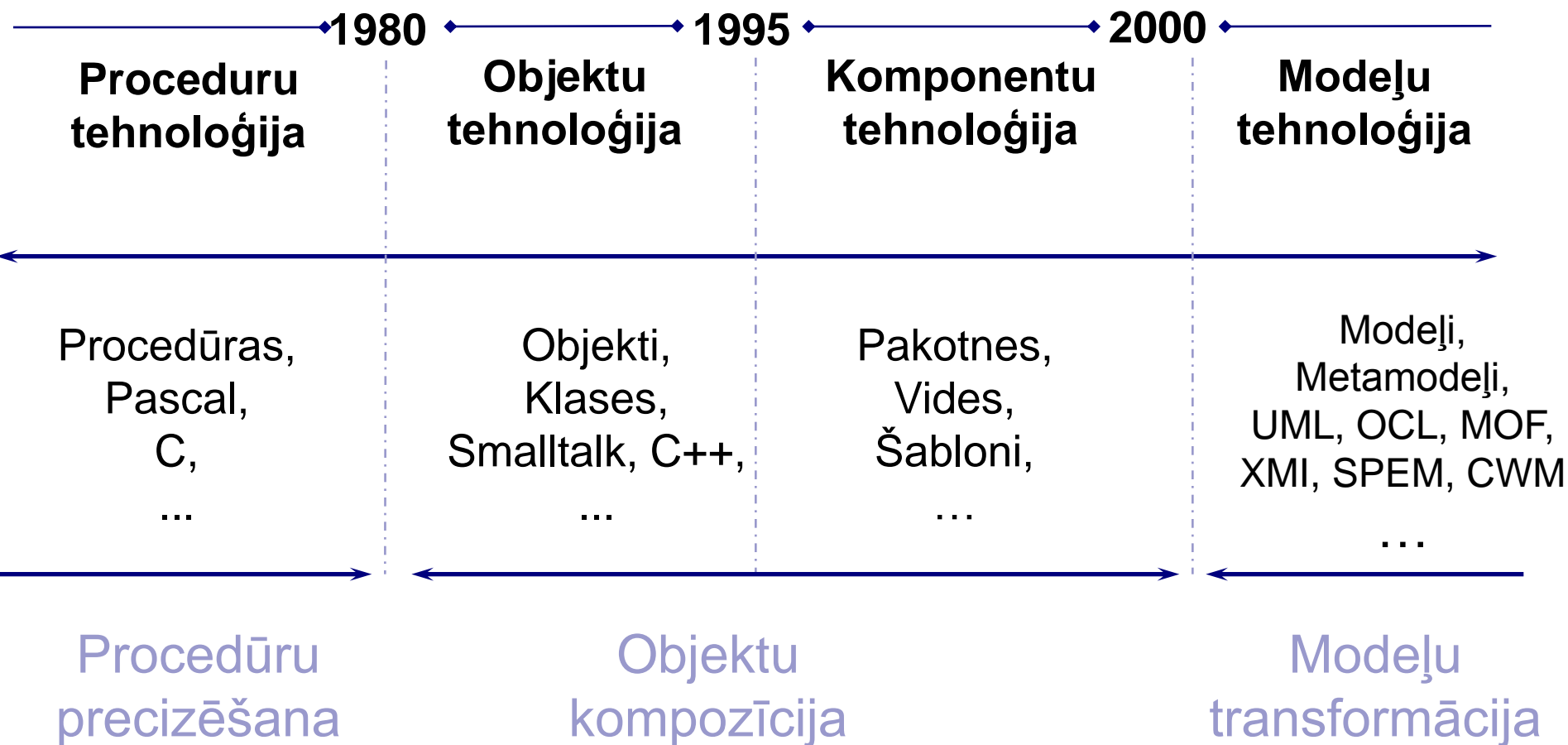
- ☐ Kas ir modelis?
- ☐ Kas ir metamodelis?
- ☐ Kas ir metametamodelis?
- ☐ Modeļa nolūki

■ Modeļvadāma arhitektūra MDA

- ☐ Modeļvadāma programmatūras izstrāde
- ☐ MDA principi

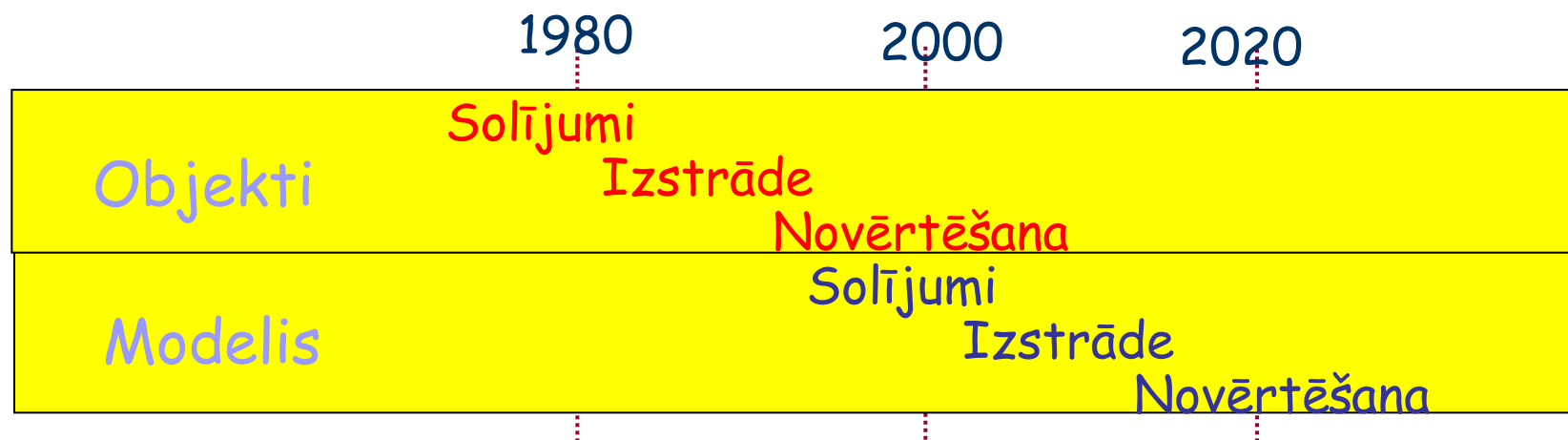
Vēsturiskais ieskats programmatūras izstrādē: tehnoloģiju attīstība

3



No objektiem uz modeļiem

- Objektu tehnoloģija realizēja apsolīto, bet nepilnīgi
 - Par iemeslu var būt tas, ka vispārināšanas meklēšana apstājas pie unifikācijas
- Pašlaik modeļvadāma inženierija (*model-driven engineering*) dod daudz solījumus
 - Vai būs iespējams tos izpildīt?
 - Izskatās, ka paliekot pie principa “viss ir modelis” tas ir iespējams



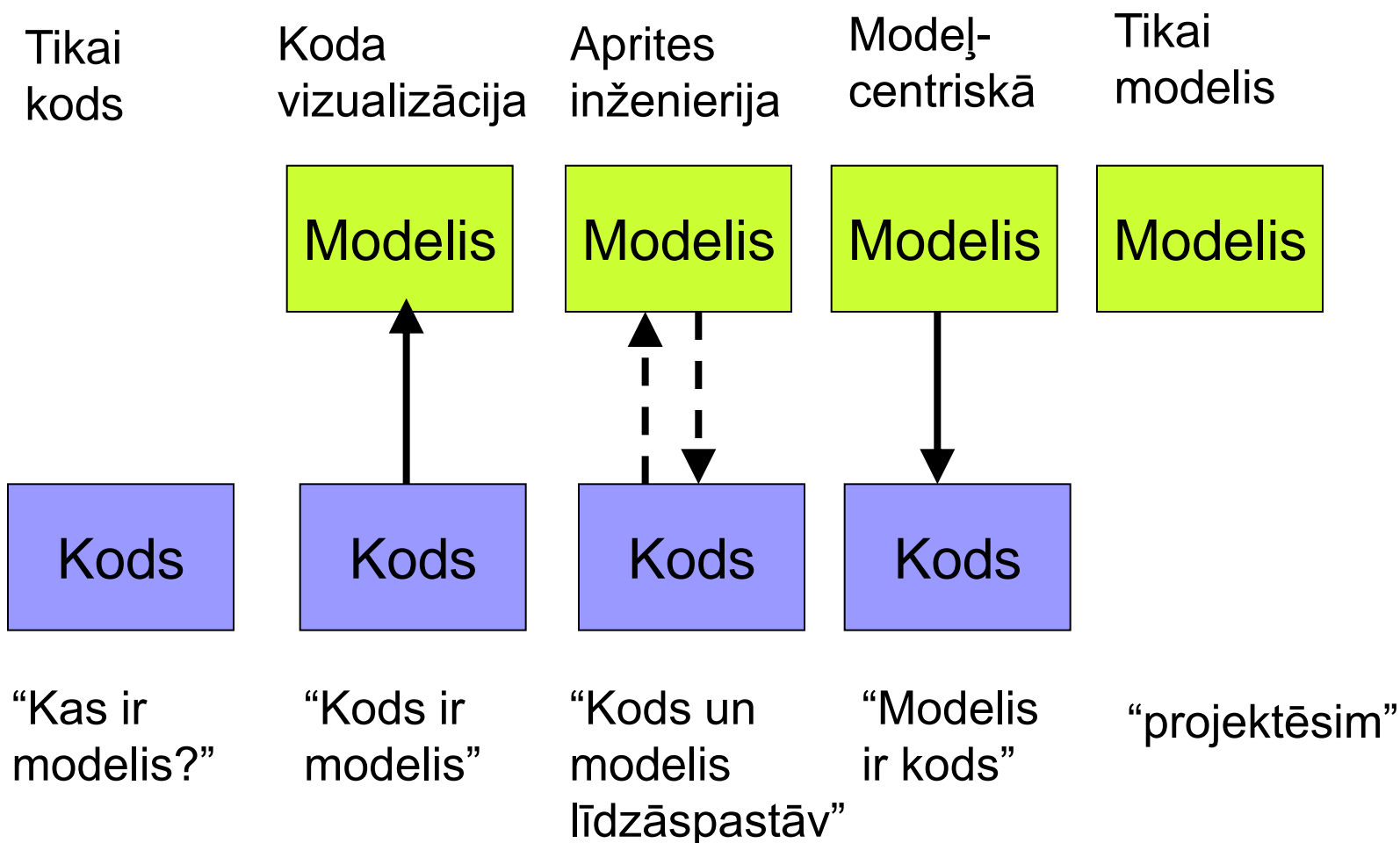
“modelī vadāms”

VS

“bāzēts uz modeļiem”

Ieskats pašreizējā programmatūras izstrādē: modelēšanas spektrs

6



leskats pašreizējā programmatūras izstrādē: modelēšanas spektrs

7

Tikai
kods

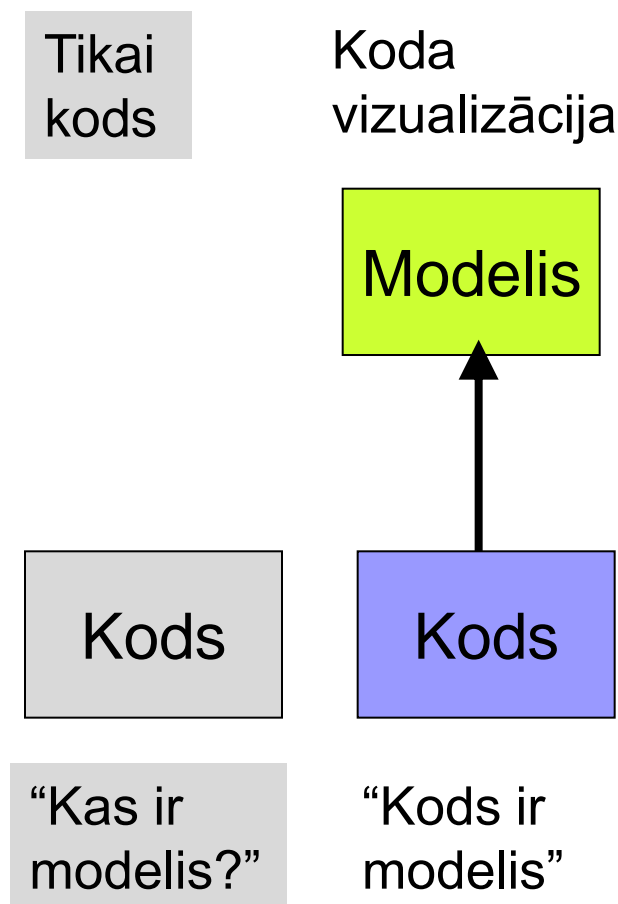
Tikai kods – vispār neizmanto atsevišķi definētus modeļus. Modeļi tiek aprakstīti uzreiz trešās paaudzes programmēšanas valodās (Java, C++, C# utt. + IDE)

Kods

“Kas ir
modelis?”

- +Individuālam darbam vai mazām komandām
- Grūti saprast sistēmas galvenos raksturojumus starp biznesloģikas detaļām
- Apgrūtināta sistēmas evolūcija

Ieskats pašreizējā programmatūras izstrādē: modelēšanas spektrs



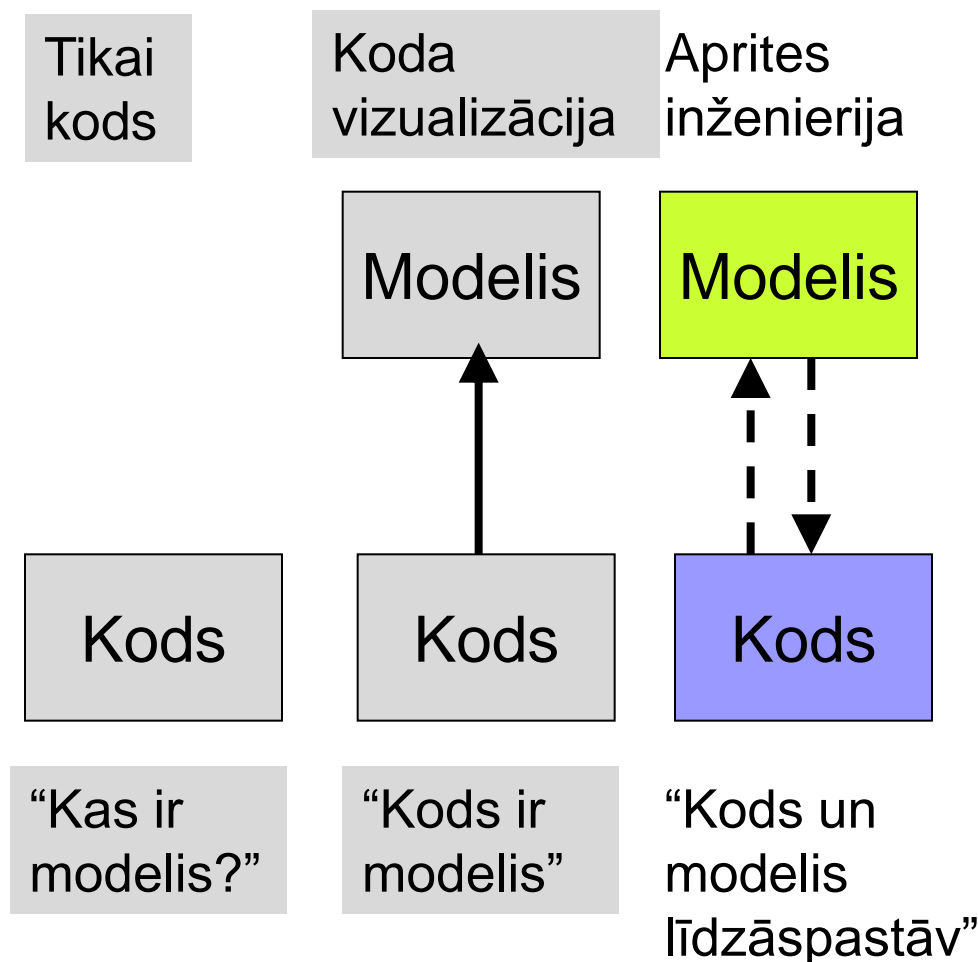
Koda vizualizācija

nodrošina koda aprakstu kādā modelēšanas notācijā. Koda (realizācijas) modelis izmainās saskaņā ar izmaiņām kodā (piemēram, kā Borland Together/J vidē).

+ Var būt izmantota koda analīzei vai grafiskai rediģēšanai

Ieskats pašreizējā programmatūras izstrādē: modelēšanas spektrs

9

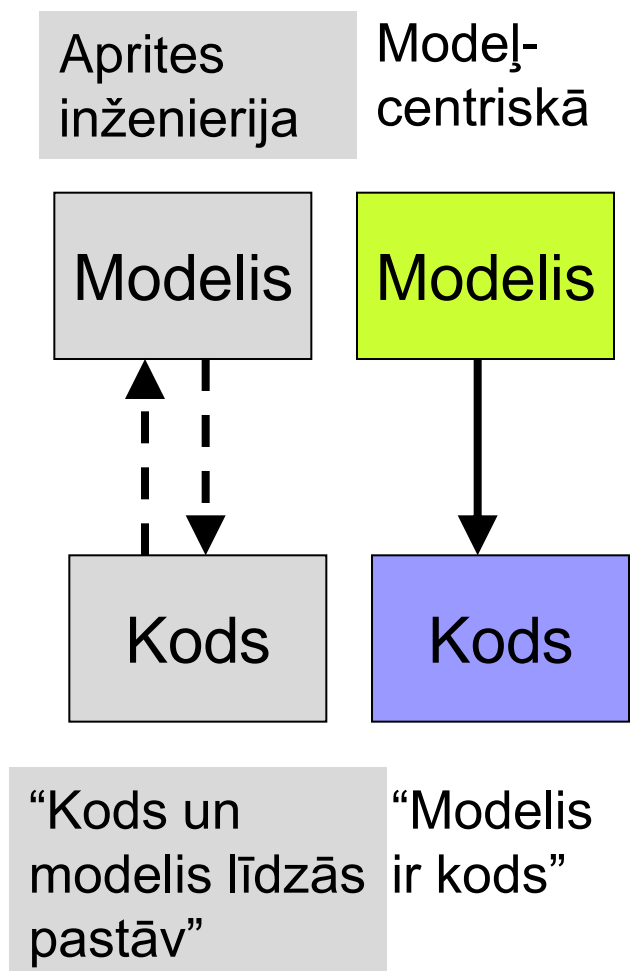


Aprite (roundtrip) inženierija

nodrošina divvirzienu apmaiņu starp abstrakto modeli un kodu. Kods tiek rakstīts balstoties uz abstrakto modeli, un modelis tiek izmainīts balstoties uz izmaiņām kodā (piemēram, IBM Rational Rose vide).

Ieskats pašreizējā programmatūras izstrādē: modelēšanas spektrs

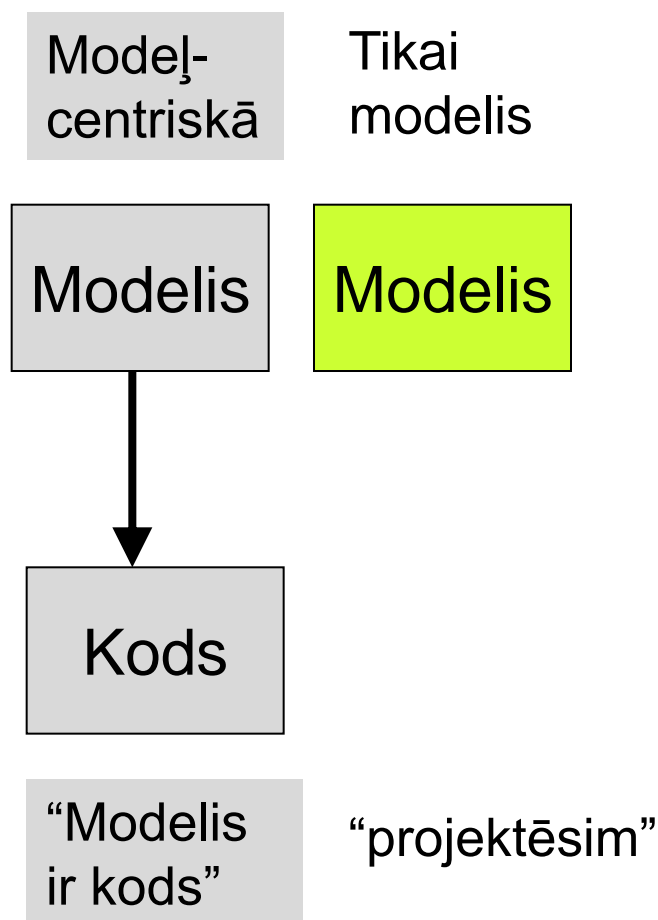
10



Uz modeļiem vērsta izstrāde – sistēmas implementācijas ģenerēšana no pietiekami detalizētiem modeļiem. Rīki, kuri pielieto šo pieeju, bieži vien atbalsta koda ģenerēšanu lietojumu noteiktiem veidiem (piemērs, IBM Rational Rose Technical Developer for real-time embedded system).

Ieskats pašreizējā programmatūras izstrādē: modelēšanas spektrs

11



Tikai modelis – šī pieeja tiek izmantota tikai lai palīdzētu izstrādātājiem vai dažādām organizācijām izprast sistēmas arhitektūru vai biznesa risinājumu. Praksē, sistēmas implementācijā var būt nesavienota ar modeļiem.

Pašreizējā izstrādes procesa vājās 12 vietas

- Nozīmīgas problēmas parādās lietojumprogrammas uzturēšanas laikā, kad prasības var izmainīties un nepieciešams veikt izmaiņas
 - Dokumentācija tiek uzturēta teksta failā vai vienkārši uz papīra
 - Specifikācijas un projektējums netiek modificēti līdz programmai
 - Prasību izmaiņas tiek veikti neformāli
 - Testi tiek izveidoti un izpildīti manuāli

Modelēšanas nepieciešamība

- Modelēšana ir dabiskā cilvēka domāšanai, tāpēc ka uzsakot katru darbību cilvēks tieši vai netieši konstruē modeli
- Asins nolaišana viduslaikos bāzējās uz nepareiza cilvēka ķermeņa modeļa
 - Tas izraisīja daudz nāves gadījumu !

Definīcijas



- **Modelis** ir **sistēmas** vienkāršots **attēls**
 - Šī īsa definīcija tiks papildināta
- Kas ir sistēma?
 - Savstarpēji saistītu elementu kopums, kas funkcionālā ziņā veido noteiktu veselumu
 - Vārda “sistēma” izcelsme ir grieķu vārds “σύστημα” [sústɛːma], kas nozīmē “sastādīts”
- Vārds “modelis” nāk no latīņu “modullus”, kas ir “modus” (mērījums) vārda pamazināmais vārds
 - Sākotnēji tas bija arhitektūras termins, kurš apzīmēja mērījumu, kas bija izmantots, lai uzstādītu dažādus attālumus starp konstruējamās ēkas daļām

Kas ir modelis?

Modelēšana, plašā kontekstā, ir kaut kā rentabla izmantošana kaut kā cita vietā kaut kādiem izziņas nolūkiem. Tas ļauj izmantot kaut ko, kas ir vienkāršāks, drošāks vai lētāks nekā realitāte (realitātes vietā) kaut kādiem nolūkiem. Modelis attēlo realitāti dotajiem nolūkiem; modelis ir realitātes abstrakcija, bet tas nevar attēlot realitātes visus aspektus. Tas ļauj rīkoties ar pasauli vienkāršotā manierē, apejot realitātes sarežģījumu, briesmas un neapgriežamību.

"The Nature of Modeling."

Jeff Rothenberg

in Artificial Intelligence, Simulation, and Modeling,

L.E. William, K.A. Loparo, N.R. Nelson, eds.

New York, John Wiley and Sons, Inc., 1989, pp. 75-92

Vārda “modelis” nozīmes

- Izcelsme - franču “modèle”, itāliešu “modella” no latīņu “modulus” (mērs, paraugs)
 - Paraugs, etalons; tips, marka.
 - Palīgsistēma (ierīce) vai matemātisko izteiksmju kopums, ko izmanto oriģināla pētīšanai; modelis aizstāj oriģinālu, saglabādams tā būtiskās īpašības.
 - Attēlojamais objekts tēlotājmākslā; persona, kas pozē mākslas fotogrāfam; persona, kas demonstrē tērpus.
 - Pamazināts, palielināts vai vienkāršots kāda priekšmeta atveidojums, piem., lidmašīnas modelis.

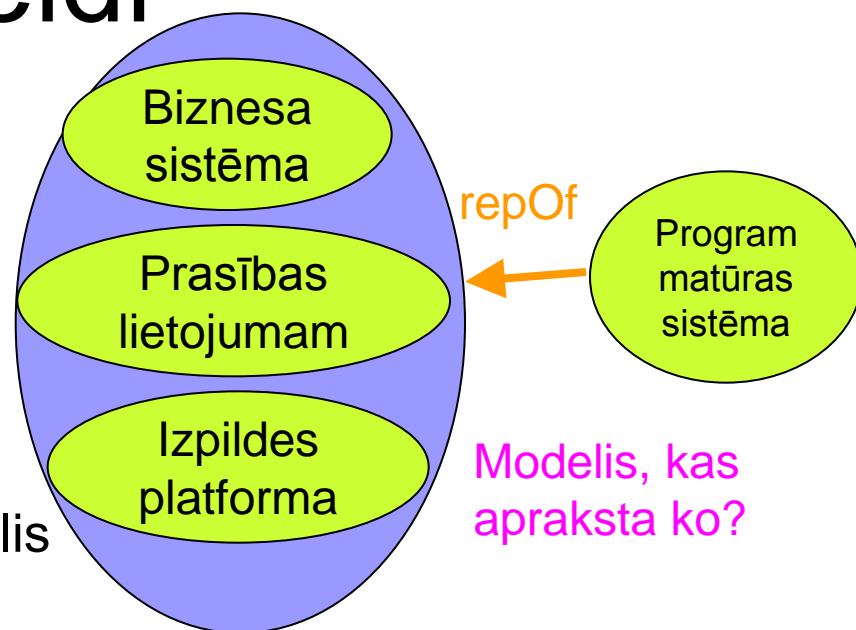
Dažādi modeļu veidi

■ Modeļu veidi

- Bioloģiskais modelis
- Ekoloģiskais modelis
- Matemātiskais modelis, utt.

■ Programmatūra kā modelis

- Tas ir sarežģīts un salikts modelis
- *Modelis, kas apraksta ko?*
 - Organizāciju, kurā programmatūra darbosies?
 - Arhitektūras, kuras programmatūra implementēs?
 - Prasības, kuras programmatūrai ir jāapmierina?
 - Komandu, kura izstrādās programmatūru?
- Kādā valodā aprakstīts?
 - Programmēšanas valodā?
 - Domēnam specifiskā valodā (DSL) !



Modelis ir atsevišķs skats uz sistēmu

Modelis A



Modelis B



Sistēma S

Modelis

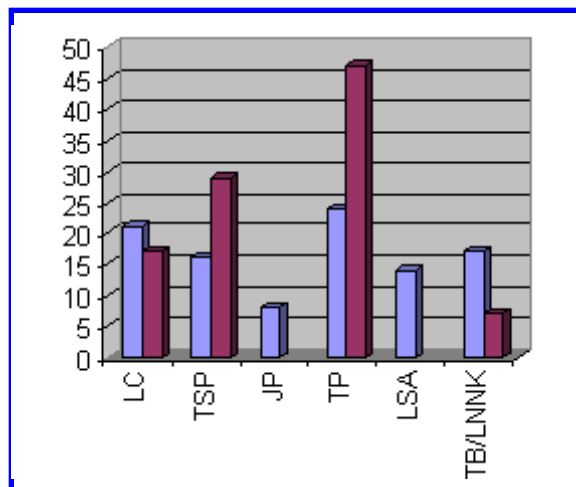
repOf

Sistēma

Modelis attēlo sistēmas dažādus aspektus

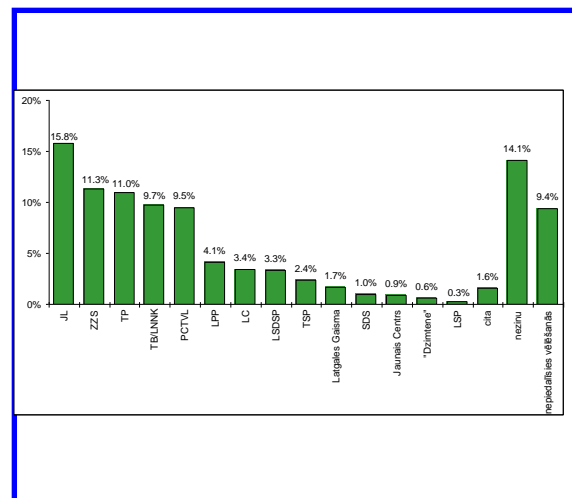
Kas ir metamodelis?

Politisko partiju sadalījums
saeimā dažādās vēlēšanu
sistēmās



■ Proportionalā vēlēšanu sistēma
■ Mažoritārā vēlēšanu sistēma

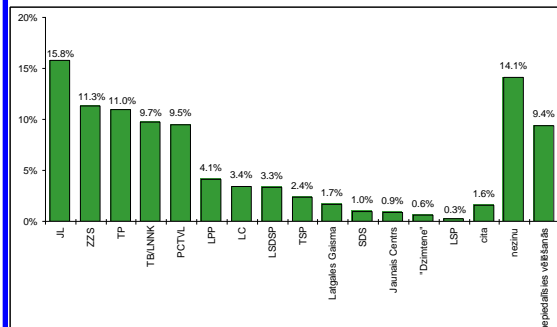
Politisko partiju reitingi
2005. gada 12.
septembrī





Netieši izteikta leģenda

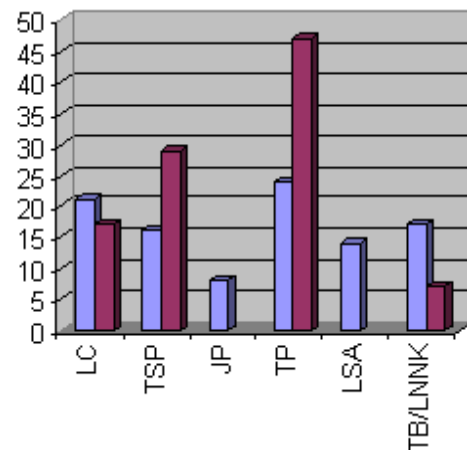
Kas ir metamodelis?

Politisko partiju sadalījums
saeimā dažādās vēlēšanu
sistēmās



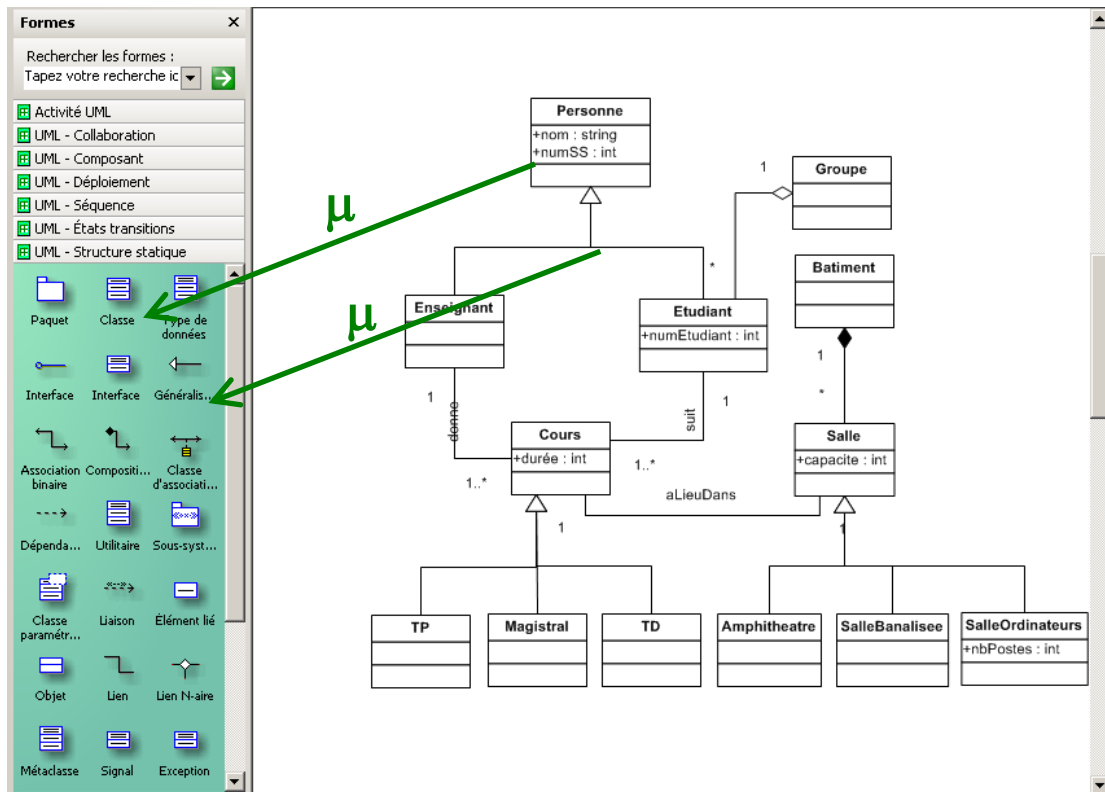
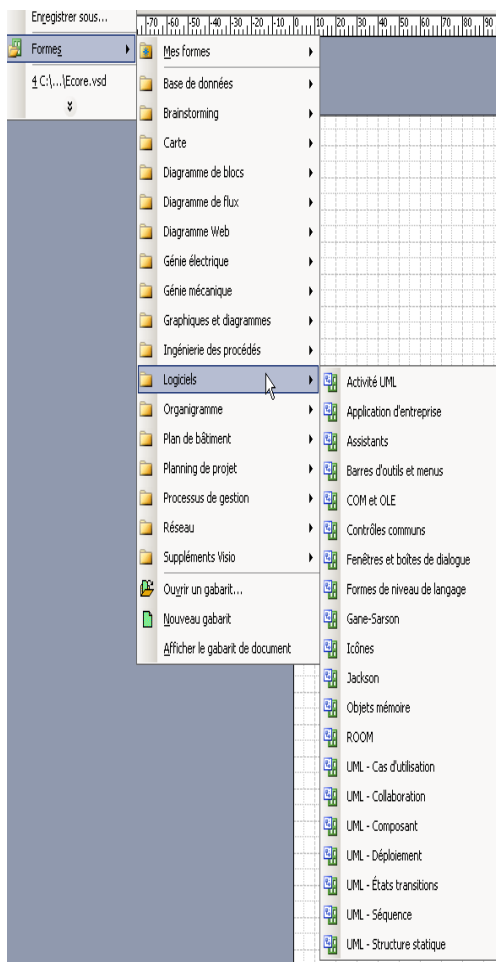
 Proporcionalā vēlēšanu sistēma
 Mažoritārā vēlēšanu sistēma

Politisko partiju reitingi
2005. gada 12.
septembrī



Netieši izteikta leģenda

Daudz šablonu dažādiem mērķiem



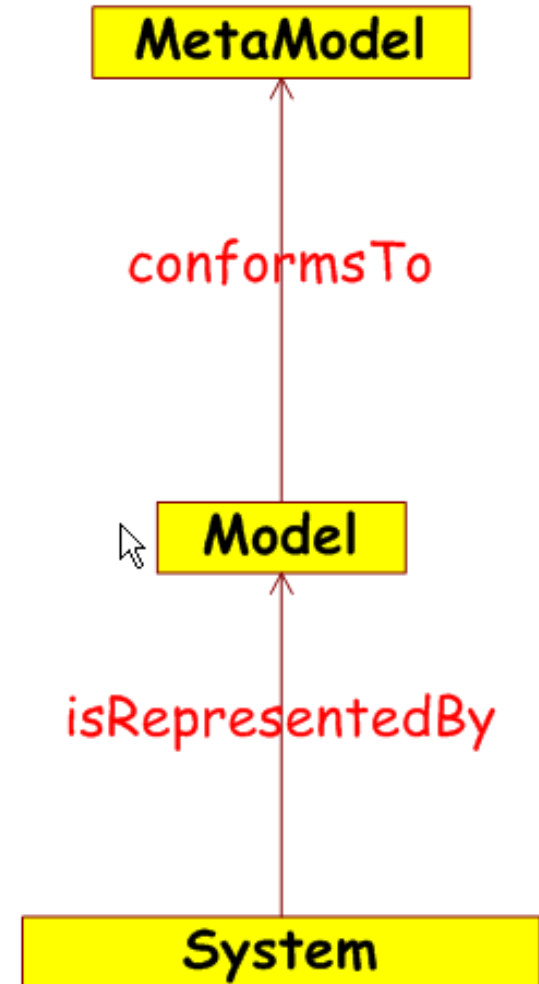
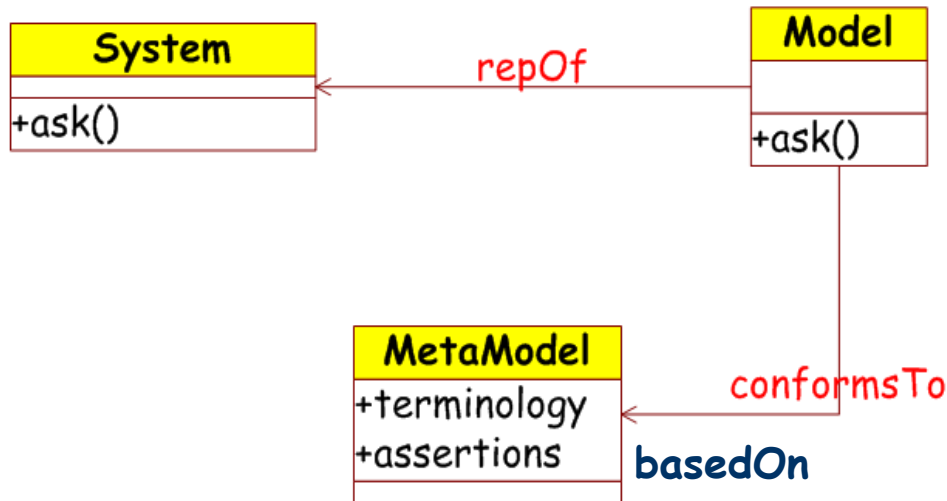
Šablonu bibliotēka

Metamodelis

c_2

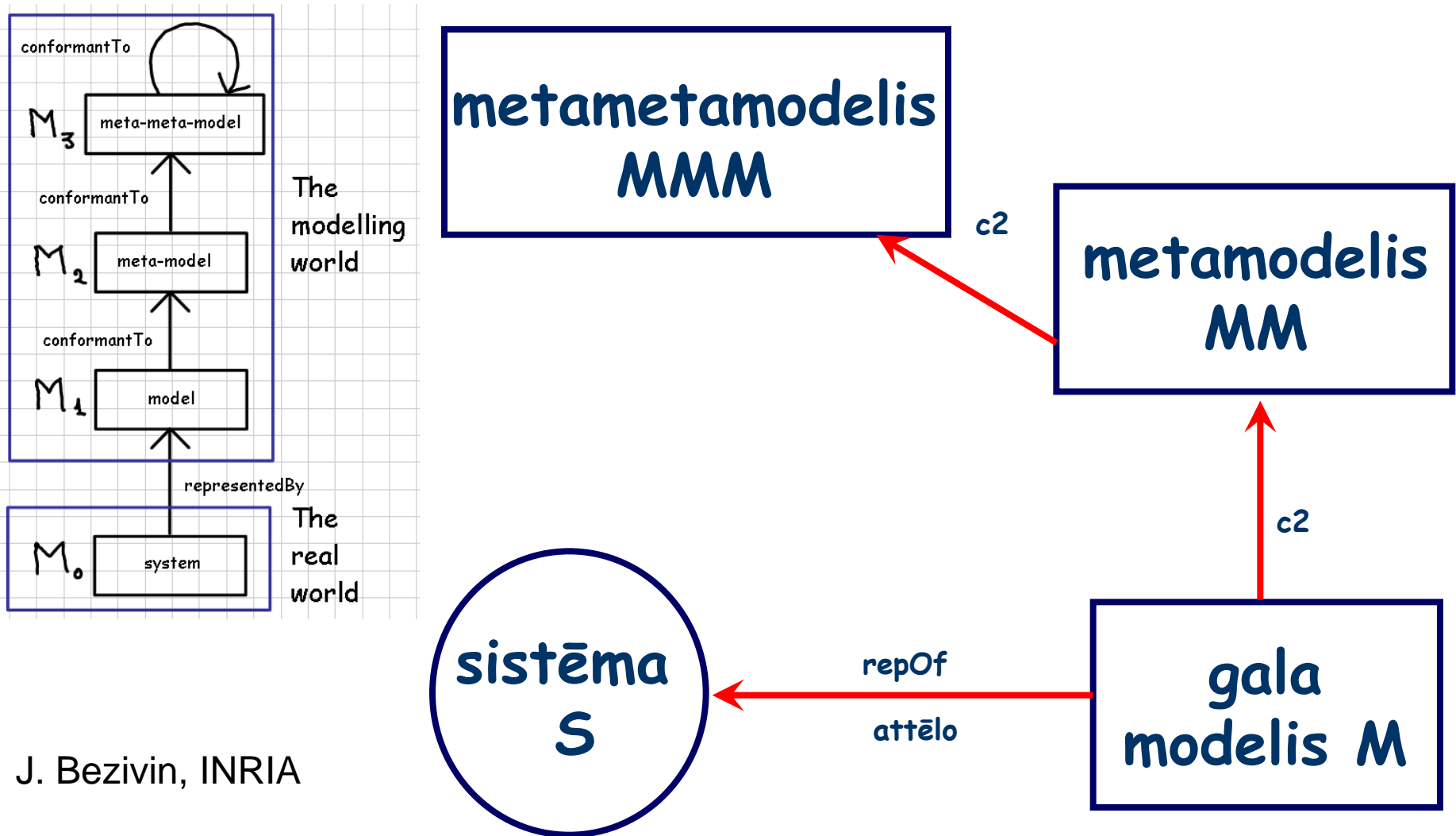
Modelis

Metamodelis ir modeļa leģenda



Sistēma, modelis, metamodelis, metametamodelis

23



Modelis

- Modelis ir kāda skatījuma uz sistēmu attēlojums
 - Sistēmu var aprakstīt vairāki modeļi
- Modelis ir uzrakstīts sava unikālā metamodela valodā
- Metamodelis ir uzrakstīts sava unikālā metametamodela valodā
- Modelis ir ierobežots orientēts iezīmēts grafs
- Modelim var būt (dažreiz) vizuālais grafiskais attēlojums

Modeļa struktūras definīcija

- **Definīcija 1.** **Orientēts multigrafs** $G = (N_G, E_G, \Gamma_G)$, kas sastāv no dažādu mezglu N_G galīgas kopas un loku E_G galīgas kopas un kartēšanas funkcijas $\Gamma_G: E_G \rightarrow N_G \times N_G$
- **Definīcija 2.** **Modelis** $M = (G, \omega, \mu)$ ir trīs elementi:
 - $G = (N_G, E_G, \Gamma_G)$ kas ir orientēts multigrafs
 - ω ir modelis, kuru sauc par M atsauču modeli un kurš ir saistīts ar grafu $G_\omega = (N_\omega, E_\omega, \Gamma_\omega)$
 - $\mu: N_G \cup E_G \rightarrow N_\omega$ ir funkcija, kura asociē G elementus (mezglus un lokus) ar G_ω elementiem (metaElementiem)

Definīcijas

- **Definīcija 3.** Metametamodelis ir modelis, kas ir atsauču modelis sev pašam (atbilst sev pašam).
- **Definīcija 4.** Metamodelis ir modelis, kura atsauču modelis ir metametamodelis.
- **Definīcija 5.** Gala modelis ir modelis, kura atsauču modelis ir metamodelis.

Lekcijas saturs

■ Modeļi

- ☐ Kas ir modelis?
- ☐ Kas ir metamodelis?
- ☐ Kas ir metametamodelis?
- ☐ Modeļa nolūki

■ Modeļvadāma arhitektūra MDA

- ☐ Modeļvadāma programmatūras izstrāde
- ☐ MDA principi

Modeļvadāmas programmatūras izstrādes (MDSD) mērķi

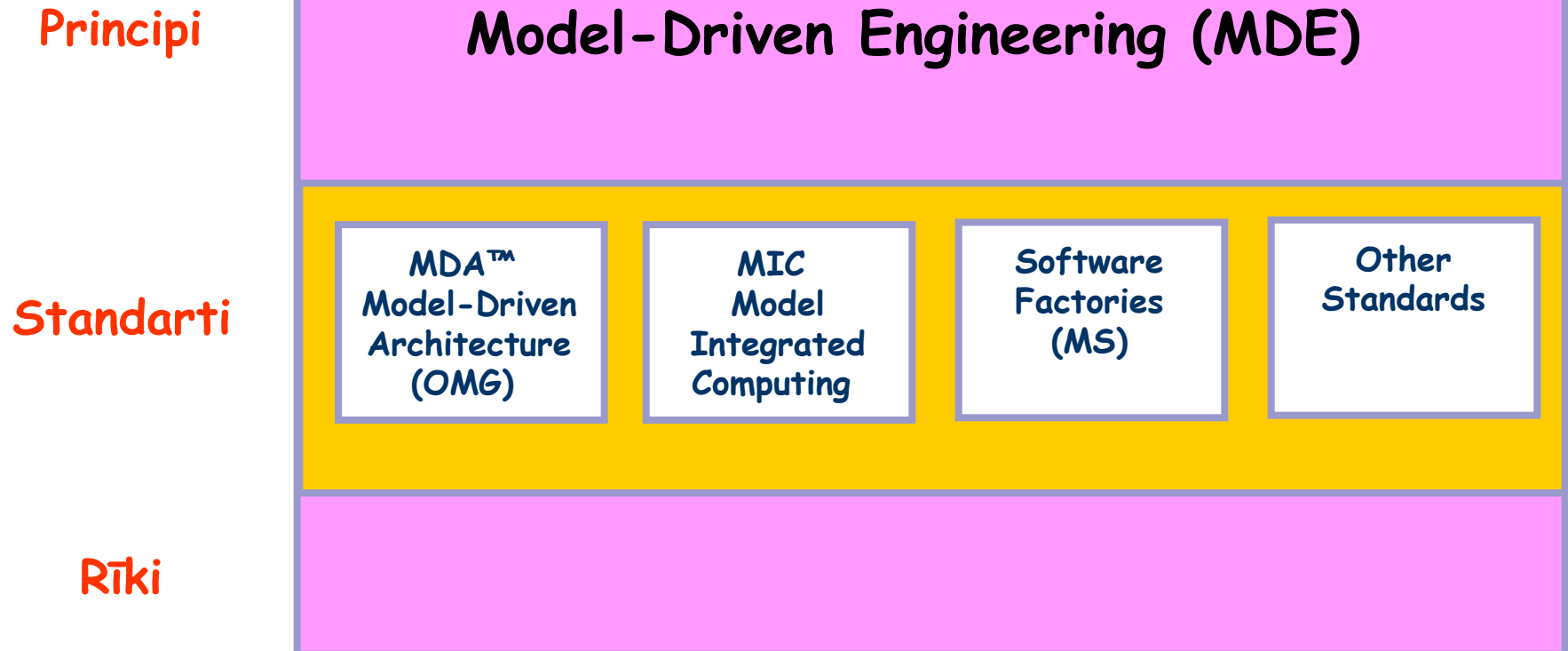
28

- Izstrādes ātruma palielināšana
- Programmatūras kvalitātes palielināšana
- Izvairīšana no dublēšanas un tehnoloģisku izmaiņu pārvaldība
- Atkārtota izmantošana
- Sarežģītības pārvaldība ar abstrakcijas palīdzību
- Ražīga vide
- Sadarbspēja un pārnesamība

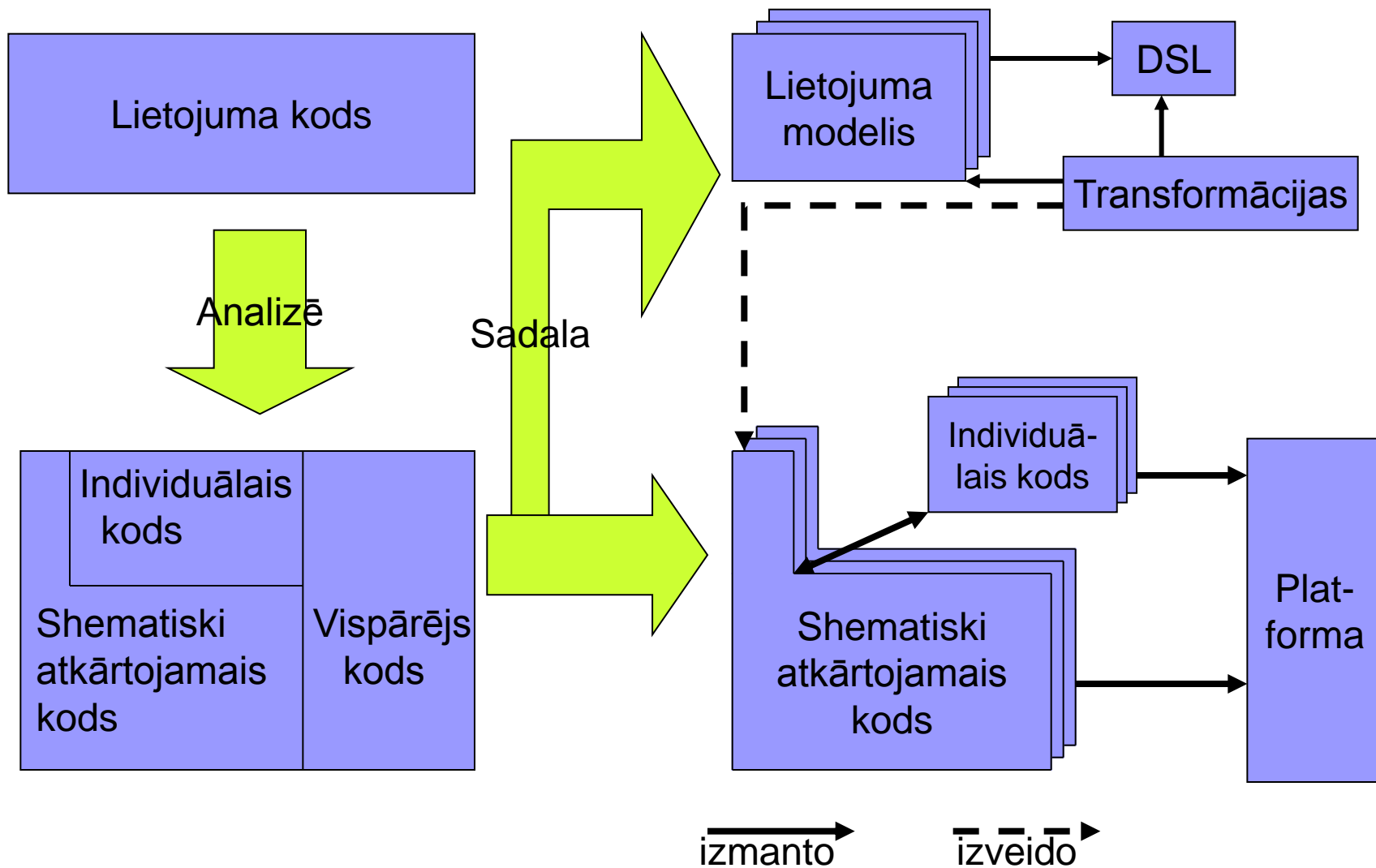
Abreviatūras

- MDE - Model Driven Engineering
- ME - Model Engineering
- MDA - Model Driven Architecture
- MDD - Model Driven Development
- MDSD - Model Driven Software Development
- MDSE - Model Driven Software Engineering
- MM - Model Management
- MDDE - Model Driven Data Engineering
- ADM - Architecture Driven Modernization
- MDRE - Model Driven Reverse Engineering
- DSL - Domain Specific Language
- DSM - Domain Specific Modeling
- Un citi patentēti nosaukumi

MDSD principi un standarti

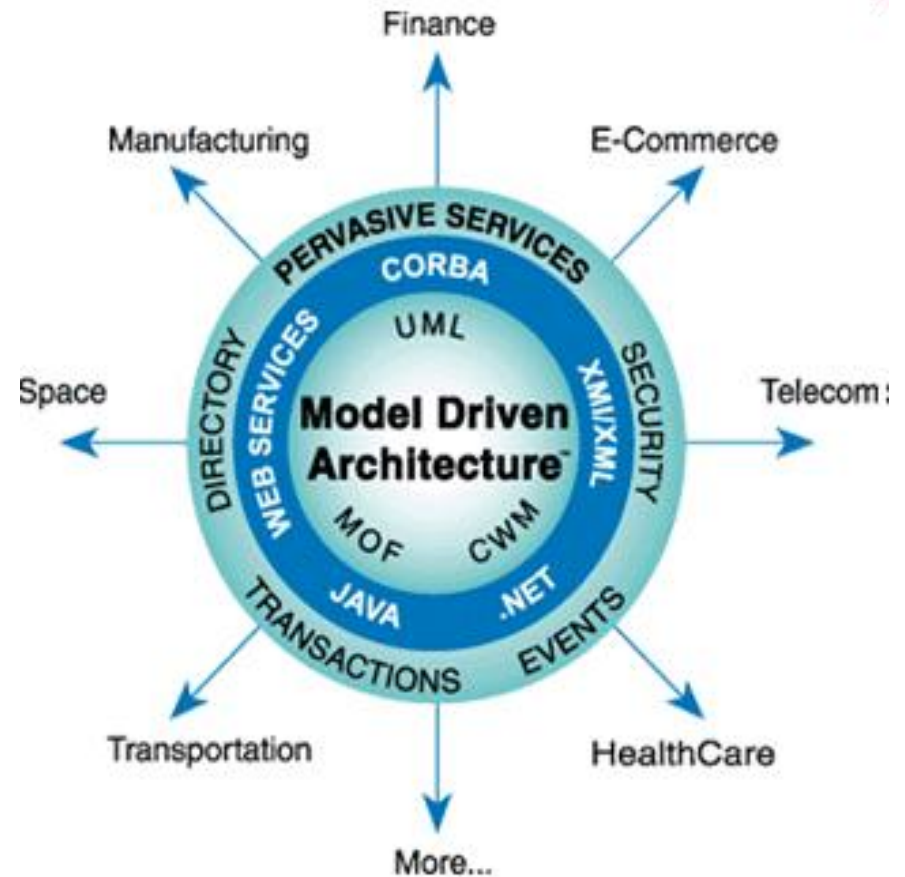


MDSD pamatideja

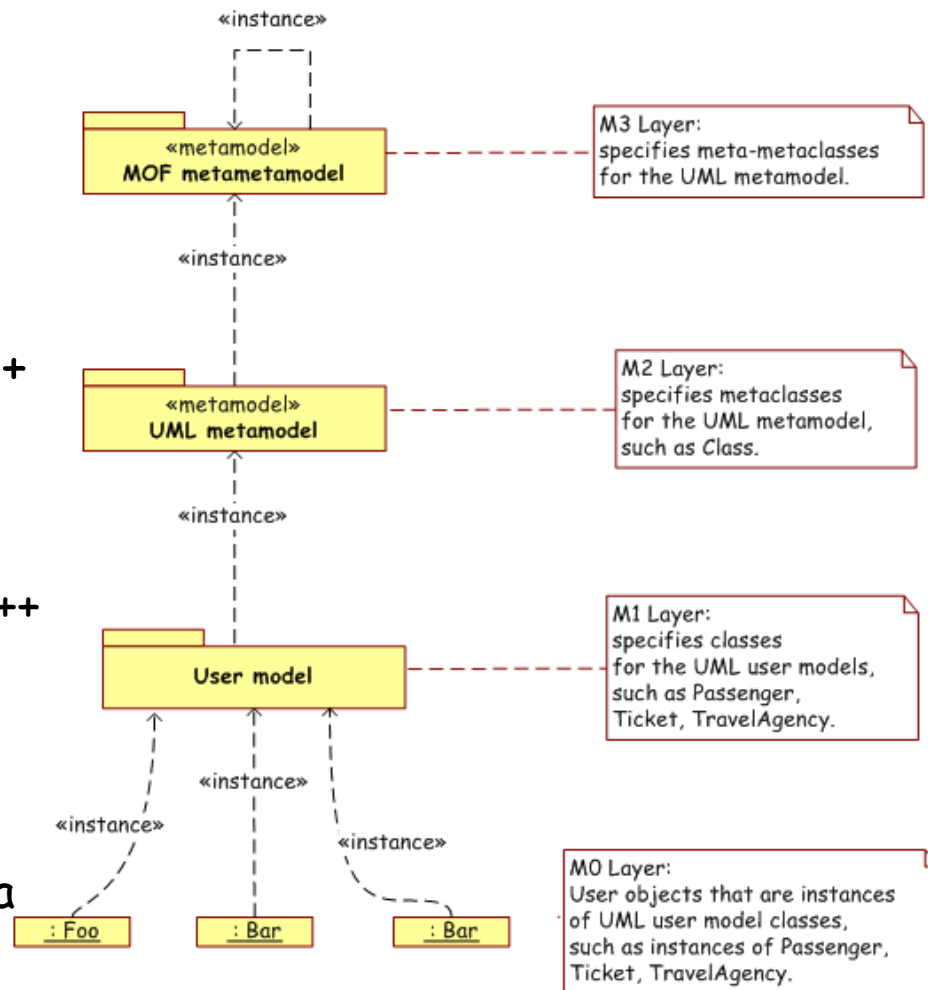
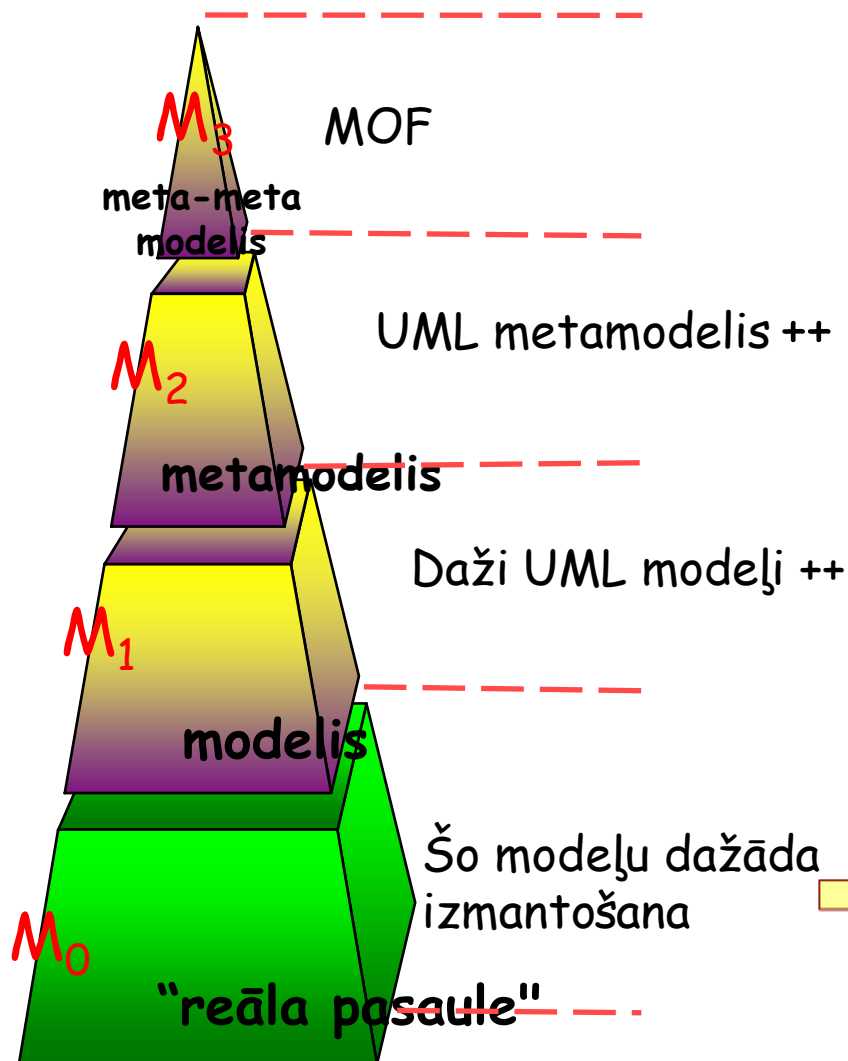


MDA mērķi un saistītie standarti

- MDA mērķi
 - Sadarbspēja
 - Pārnesamība
 - Atkārtota izmantošana
- Standarti
 - Meta-Object Facility (MOF)
 - Unified Modeling Language (UML)
 - Common Warehouse Metamodel (CWM)



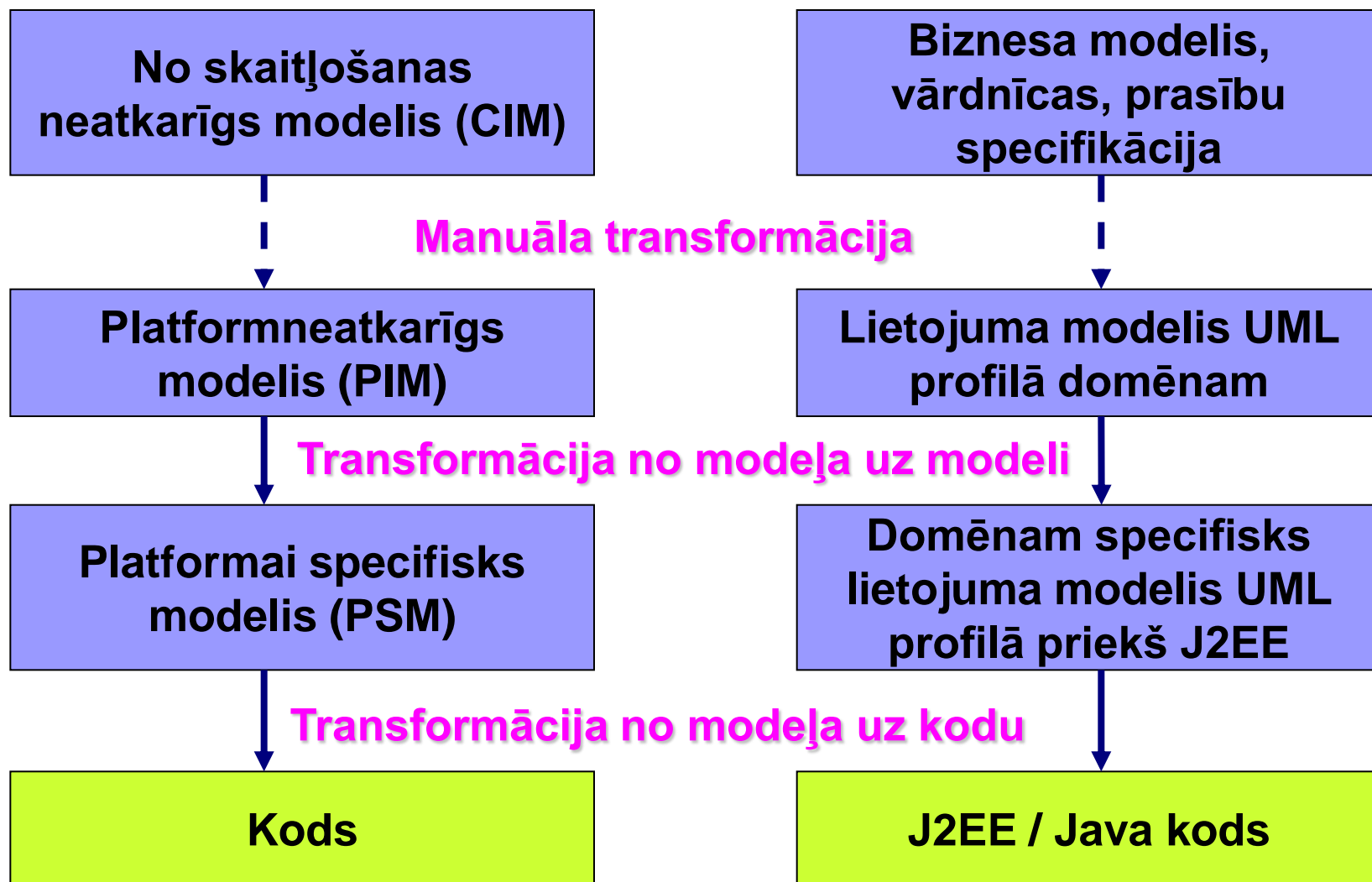
OMG grupas MDA steks



Modeļvadāmas arhitektūras terminoloģija

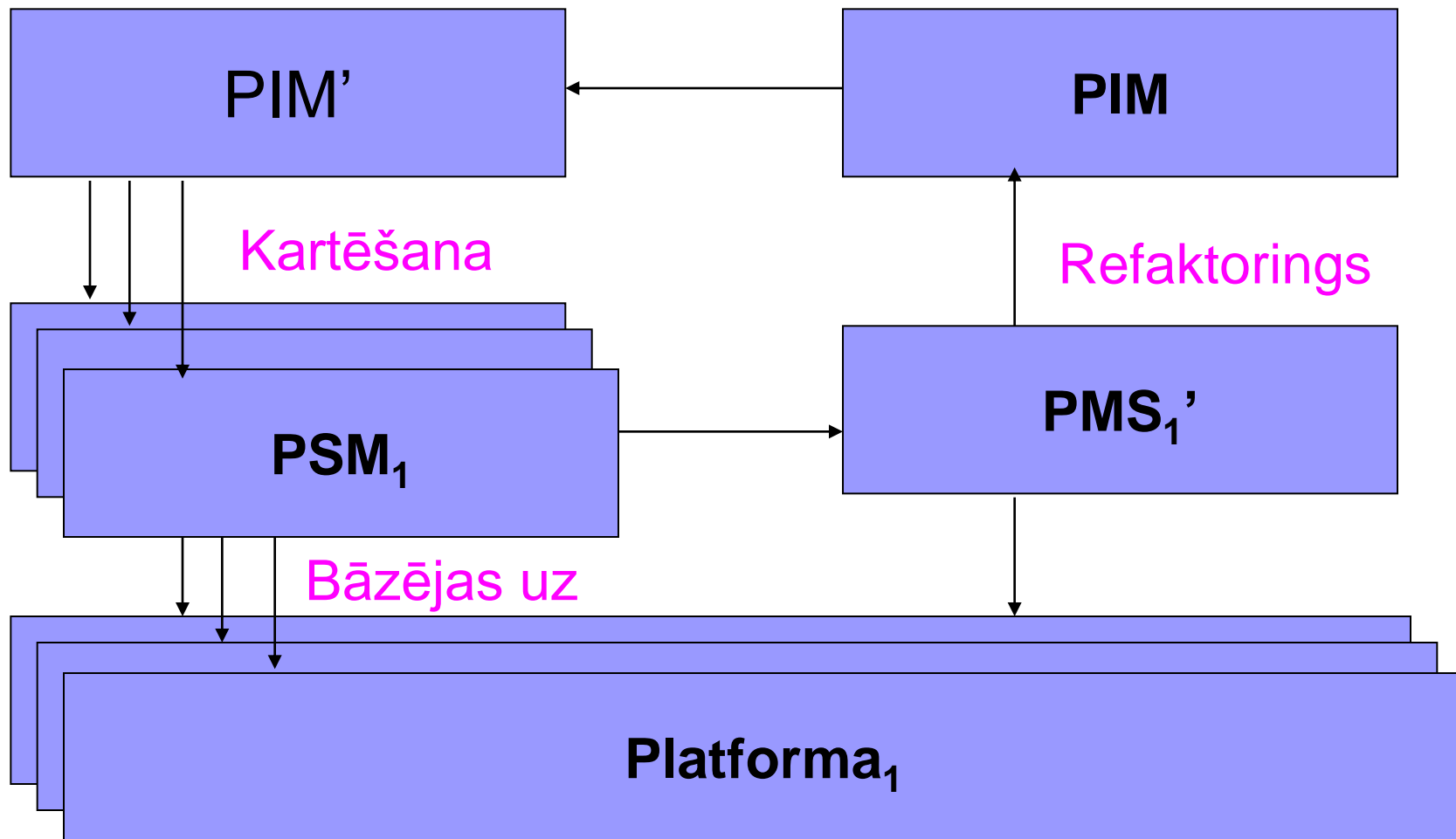
- **MDA skatupunkti uz sistēmu**
 - No skaitļošanas neatkarīgs
 - Platformneatkarīgs
 - Platformai specifisks
- **Modelis** ir sistēmas struktūras, funkcionēšanas vai uzvedības abstrakts attēlojums
- **Platforma** – MDA neko nesaka par platformas abstrakcijas līmeni. Atkarībā no konteksta par platformu var būt CORBA, J2EE, Web servisi, lietojuma arhitektūra utt.
- **Transformācijas** attēlo modeļus nākamajā līmenī (tas var būt modelis vai avota kods)
 - M2 (*model-to-model*) transformācijas pārveido kādu modeli citā
 - *Model-to-Code* transformācijas pārveido kādu modeli kodā

Modeļvadāmas arhitektūras (Model-Driven Architecture, MDA) koncepcijas



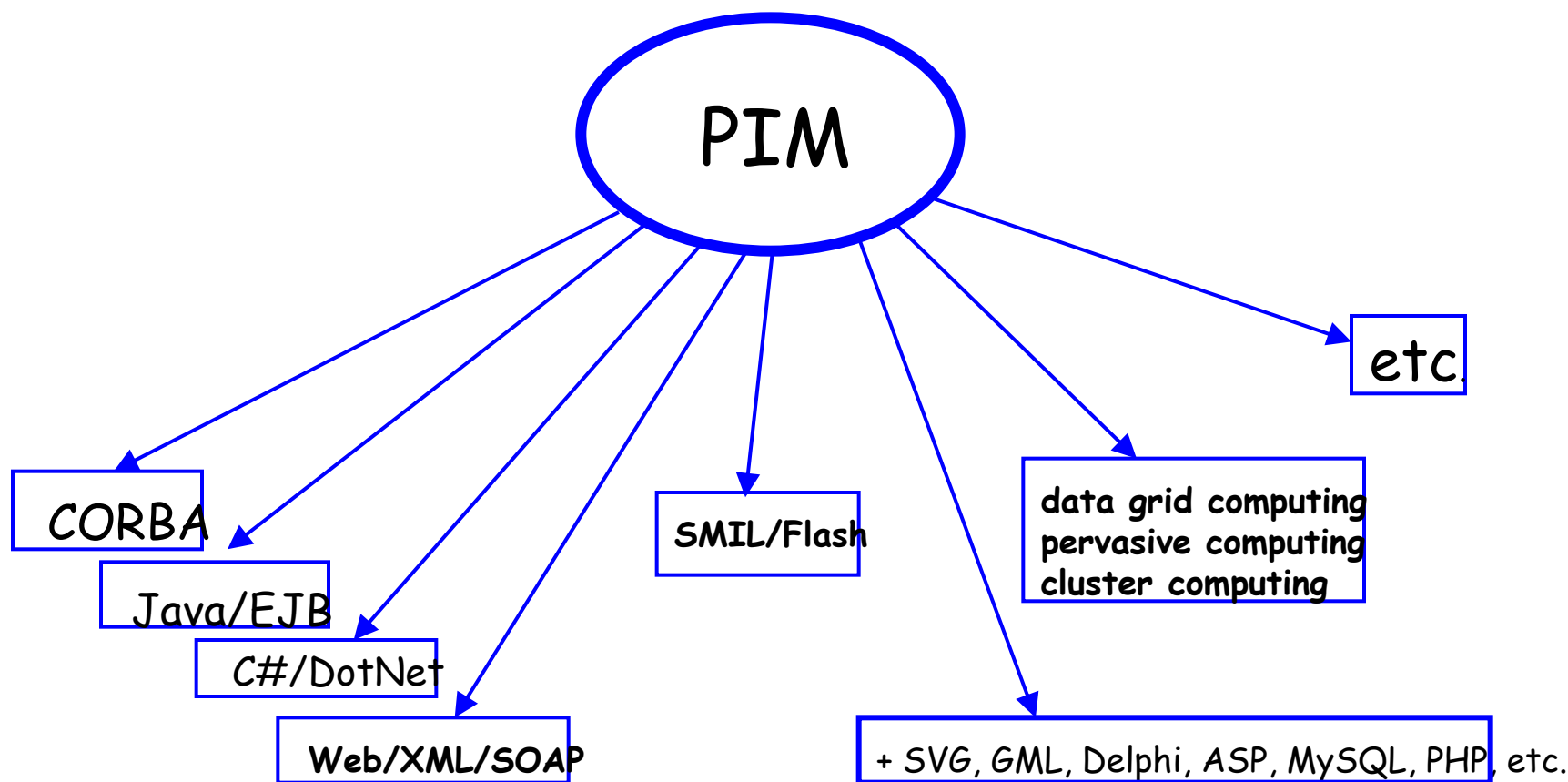
Modeļvadāmas arhitektūras terminoloģija: PIM un PSM

36



~~Vienu reizi uzraksti, palaiž visur~~ 37

Vienu reizi modelē, ģenerē visur



Programmatūras izstrāde ir modeļu transformāciju virkne

