

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
VERSLO INFORMATIKOS KATEDRA

Saulius Bekeris

**Objektinių modelių transformacijų XSLT ir QVT  
palyginimas**

Magistro darbas

Darbo vadovas  
doc. V. Pilkauskas

Kaunas, 2005

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
VERSLO INFORMATIKOS KATEDRA

TVIRTINU  
Katedros vedėjas  
prof. habil. dr. H. Pranevičius  
2005 05

## **Objektinių modelių transformacijų XSLT ir QVT palyginimas**

Informatikos mokslo magistro baigiamasis darbas

Kalbos konsultantė  
Lietuvių k. katedros lektorė  
dr. J. Mikelionienė  
2005 05

Vadovas  
doc. dr. V. Pilkauskas  
2005 05

Recenzentas  
dr. doc. Rimantas Butleris  
2005 05

Atliko  
IFM-9/1 gr. stud.  
S. Bekeris  
2005 05

Kaunas, 2005

# TURINYS

SUMMARY .....	6
1. ĮVADAS .....	7
2. MDA ARCHITEKTŪRA .....	11
3. AGREGATINĖS SISTEMOS .....	15
4. XSLT.....	27
4.1. XSLT transformacijos.....	27
4.2. Modelių transformacijos .....	27
4.2.1. Elemento Operacija transformacija.....	28
4.2.2. Elemento Sujungimas transformacija .....	28
4.2.3. Elemento Pranesimas transformacija .....	29
4.2.4. Elemento Eile transformacija.....	29
4.2.5. Elemento Busena transformacija .....	30
4.2.6. Elemento Agregatas transformacija.....	30
4.3. XSLT transformacijos apibendrinimas .....	31
5. QVT .....	32
5.1. QVT standartas .....	32
5.2. Transformacijos .....	32
5.3. Modelių transformacijos .....	35
5.3.1. Elemento Operacija transformacija.....	36
5.3.2. Elemento Sujungimas transformacija .....	37
5.3.3. Elemento Pranesimas transformacija .....	38
5.3.4. Elemento Eile transformacija.....	38
5.3.5. Elemento Busena transformacija .....	39
5.3.6. Elemento Agregatas transformacija.....	40
5.4. QVT transformacijos apibendrinimas.....	41
6. IŠVADOS .....	42
7. LITERATŪRA .....	44
8. TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	45
9. PRIEDAI.....	46
9.1. PIM metamodelio aprašas XMI formatu .....	46
9.2. PIM modelio aprašas XML formatu .....	47
9.3. PSM metamodelio aprašas XMI formatu .....	48
9.4. Java kodo atitinkančio PSM modelio klasę Agregatas1 pavyzdys.....	50
9.5. XSLT transformacija .....	51
9.6. PSM modelio aprašas XML formatu .....	55
9.7. QVT transformacijos aprašas.....	58

## LENTELĒS

1 lent. Elemento Operacija transformacija.....	28
2 lent. Elemento Sujungimas transformacija .....	28
3 lent. Elemento Sujungimas transformacija .....	29
4 lent. Elemento Sujungimas transformacija .....	29
5 lent. Elemento Sujungimas transformacija .....	30
6 lent. Elemento Sujungimas transformacija .....	30
7 lent. Terminai .....	45
8 lent. Santrumpos .....	45

## PAVEIKSLAI

1. pav. Tradicinis programinės įrangos kūrimo ciklas.....	8
2. pav. MDA programinės įrangos kūrimo procesas .....	9
3. pav. Pagrindiniai etapai MDA programinės įrangos kūrimo procese.....	12
4. pav. MDA sistemų tarpusavio veikimas naudojant jungiamuosius ryšius .....	13
5. pav. Agregatinės sistemos modelis .....	15
6. pav. PIM ir PSM metamodeliai .....	16
7. pav. PIM metamodelis .....	17
8. pav. PIM metamodelis Eclipse aplinkoje.....	18
9. pav. PIM modelis Eclipse aplinkoje .....	18
10. pav. PSM metamodelis .....	20
11. pav. PSM metamodelis Eclipse aplinkoje.....	21
12. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas1 klasė.....	22
13. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas2 klasė.....	23
14. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas3 klasė.....	24
15. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas4 klasė.....	25
16. pav. Modeliavimo biblioteka. ....	25
17. pav. Bendras XSLT naudojimo atvejis .....	27
18. pav. Žymėjimai naudojami QVT standarto pasiūlyme .....	33
19. pav. Aukšto lygio ryšis ir jo realizacija .....	33
20. pav. Transformacijos, ryšiai ir jų realizacijos .....	34
21. pav. Ryšio aprašymo forma .....	35
22. pav. Ryšio realizacijos aprašymo forma .....	35
23. pav. Ryšio relationOperacija grafinė forma.....	36
24. pav. Ryšio relationOperacija tekstinė forma.....	36
25. pav. Ryšio realizacijos mappingOperacija tekstinė forma.....	37
26. pav. Ryšio relationSujungimas grafinė forma .....	37
27. pav. Ryšio relationSujungimas tekstinė forma .....	37
28. pav. Ryšio realizacijos mappingSujungimas tekstinė forma .....	37
29. pav. Ryšio relationPranesimas grafinė forma .....	38
30. pav. Ryšio relationPranesimas tekstinė forma.....	38
31. pav. Ryšio realizacijos mappingPranesimas tekstinė forma.....	38
32. pav. Ryšio relationEile grafinė forma .....	39
33. pav. Ryšio relationEile tekstinė forma.....	39
34. pav. Ryšio realizacijos mappingEile tekstinė forma.....	39
35. pav. Ryšio relationBusena grafinė forma.....	39
36. pav. Ryšio relationBusena tekstinė forma .....	40
37. pav. Ryšio realizacijos mappingBusena tekstinė forma .....	40
38. pav. Ryšio relationAgregatas grafinė forma .....	41

## SUMMARY

This work analysis model transformation problems in Model Driven Architecture (MDA). MDA is an architecture created by Object Management Group (OMG). MDA separates business and application logic from underlying platform technology. Platform-independent models of an application or system can be realized through the MDA on any platform.

The most important MDA problem is a transformation between platform-independent models (PIM) and platform specific models (PSM). Currently, there is no standard language for these transformations. Purpose of this work is to analyze possible transformations.

Two technologies will be researched: XSLT and QVT

XSLT (Extensible Stylesheet Language Transformations) is a language for transforming XML documents into other XML documents.

QVT (Queries/Views/Transformations) is a language for transforming models in MDA architecture.

Aggregation systems will be used as models for transformations. Platform-independent model of aggregation system will be created and transformed to Java platform models using XSLT and QVT transformations.

# 1. ĮVADAS

Progresas programinės įrangos kūrimo srityje dažnai yra lyginamas su kompiuterių techninės įrangos gamyba. Kol pastaruosius dvidešimt metų techninė įranga sparčiai tobulėjo (pvz. procesoriaus greitis didėjo eksponentiškai), progresas programinės įrangos kūrimo srityje atrodo yra minimalus. Taip yra todėl kad progresas programinės įrangos kūrime negali būti matuojamas programavimo greičiu ar kaina.

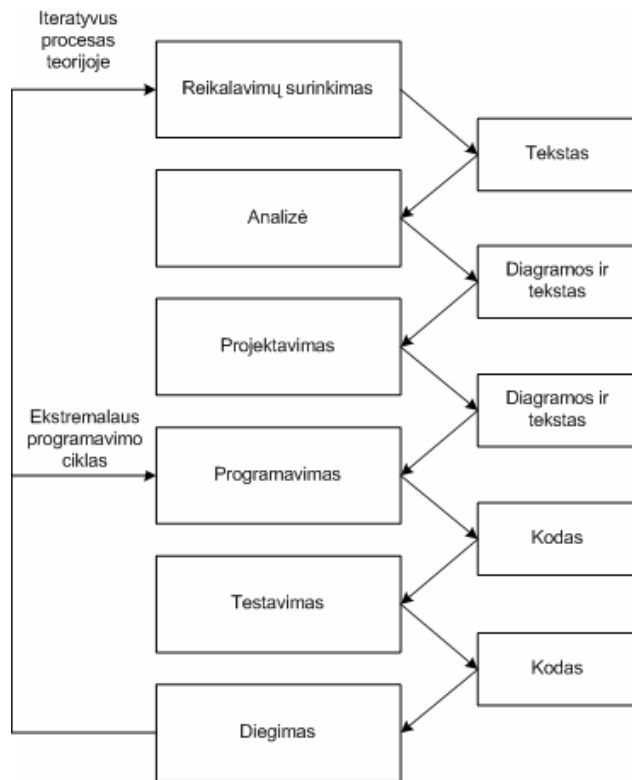
Progresas programinės įrangos kūrimo srityje yra akivaizdus pagal faktus kad įmanoma sukurti vis sudėtingesnes ir didesnes sistemas. Sukurti paprastas sistemas, kurios neturi grafinės vartotojo sąsajos ir nėra sujungtos su kitomis sistemomis šiais laikais galima labai greitai. Tačiau tokios sistemos jau nebėra kuriamos.

Vis dar programinės įrangos kūrimas yra sritis, kurioje sutinkama daug problemų. Programinės įrangos rašymas užima daug laiko. Kiekvieną kartą atsiradus naujai technologijai turi būti įdedama daug darbo jos įsisavinimui ir pritaikymui. Sistemos nėra kuriamos naudojant tik vieną technologiją ir joms visada tenka bendrauti su kitomis sistemomis. Taip pat kuriant programinę įrangą susiduriama su problemomis dėl dažnai besikeičiančių reikalavimų.

Modeliais pagrįsta architektūra MDA (*Model Driven Architecture*) yra programinės įrangos kūrimo architektūra sukurtas OMG (*Object Management Group*). Pagrindinis MDA aspektas yra modelių svarba programinės įrangos kūrimo procese. Naudojant MDA programinės įrangos kūrimo procesas yra pagrįstas sistemos modelių kūrimu. MDA architektūra bando spręsti problemas sutinkamas tradiciniame programinės įrangos kūrimo procese.

Viena problema yra programinės įrangos kūrimo produktyvumas. Programinės įrangos kūrimo procesas dažnai yra paremtas žemo lygio projektavimu ir programavimu. Tipiškas procesas susideda iš šių fazių:

1. Reikalavimų surinkimas
2. Analizė ir funkcionalumo aprašymas
3. Projektavimas
4. Programavimas
5. Testavimas
6. Diegimas



1. pav. Tradicinis programinės įrangos kūrimo ciklas

Dokumentacija šiame procese yra rašoma tik pirmose trijose fazėse. Prasidėjus programos rašymo fazei atsiranda skirtumai tarp dokumentacijos ir realaus programos realizavimo. Jei dokumentacija nėra nuolat atnaujinama ji nebeatitinka programos kodo. Kai sistema yra keičiama laikas nuo laiko, šis skirtumas vis didėja. Pakeitimai dažnai yra daromi tik programos kode, nes laiko diagramų ir kitos dokumentacijos atnaujinimui dažnai nebūna.

Kita problema programinės įrangos kūrimo srityje yra portatyvumas. Kiekvienais metais naujos technologijos yra sukuriamos ir tampa populiaros. Programų kūrėjams tenka pereiti prie naujų technologijų, nes jos dažnai išsprendžia esamas problemas, to reikalauja vartotojai arba tiesiog programų kūrimo priemonės nebedirba su senomis technologijomis. Programuotojams tenka pereiti prie naujų technologijų, todėl investicijos į senas technologijas praranda savo vertę.

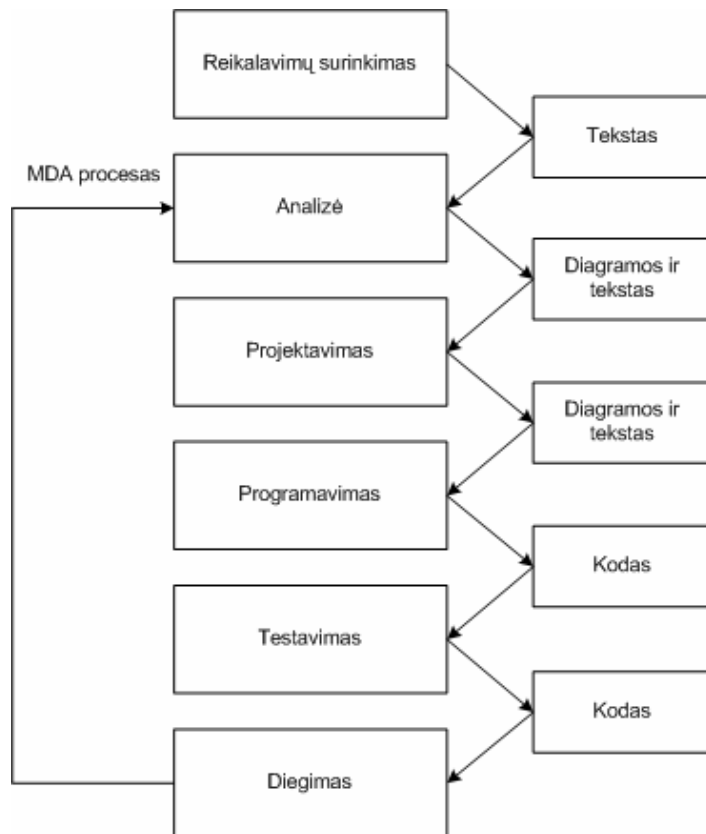
Taip pat yra sutinkama programų tarpusavio veikimo problema. Sistemos retai veikia atskirai. Daugeliui sistemų reikia bendrauti su kitomis, dažnai jau egzistuojančiomis sistemomis parašytomis įvairiausiomis technologijomis. Net kai yra kuriamos naujos sistemos, jos dažnai apima tiek naujas tiek senas technologijas.



Dokumentacijos rašymas ir jos atnaujinimas visada buvo programinės įrangos kūrimo proceso problema. Dokumentacijos rašymas užima daug laiko, ji nėra susieta su programos kodu, todėl visus pakeitimus, kurie atliekami kode, reikia padaryti ir dokumentacijoje. Šios problemos sprendimas yra programos, generuojančios dokumentaciją iš programos kodo. Taip yra užtikrinama, kad dokumentacija visada atitinka programos kodą ir yra sutaupoma laiko. Tačiau tai išsprendžia tik žemo lygio dokumentacijos problemą. Aukštesnio lygio dokumentacija (tekstas ir diagramos) turi būti kuriama rankiniu būdu. Kadangi kuriamos sistemos yra pakankamai sudėtingos, tai aukštesnio abstrakcijos lygio dokumentacija yra būtina.

Šiame darbe nagrinėjamos modeliais pagrįstos architektūros MDA (*Model Driven Architecture*) modelių transformacijų problemos. MDA – tai OMG grupės (*Object Management Group*) sukurta architektūra.

MDA programinės įrangos kūrimo ciklas, kuris yra parodytas žemiau esnačiame paveiksle turi tas pačias fazes kaip ir tradicinis ciklas. Pagrindinis skirtumas yra rezultatas gaunamas programinės įrangos kūrimo proceso metu. Gaunami formalūs modeliai kurie gali būti suprantami kompiuterių.



2. pav. MDA programinės įrangos kūrimo procesas

Pagrindiniai MDA architektūros elementai yra šie modeliai:

- Nuo platformos nepriklausantis modelis PIM (*Platform Independent Model*)
- Nuo platformos priklausantis modelis PSM (*Platform Specific Model*)

Didžiausia problema MDA architektūroje šiuo metu yra PIM modelio transformacija į PSM modelį. Šis etapas dar nėra standartizuotas. Šio darbo tikslas yra panagrinėti galimus modelių transformacijos būdus.

Viena iš technologijų galinčių atlikti šią transformaciją yra XSLT. XSLT yra kalba skirta transformuoti XML dokumentus į kitus XML dokumentus. Kadangi standartinis modelių saugojimo formatas MDA architektūroje yra XMI (*XML Metadata Interchange*), XSLT kalba bus bandoma transformuoti vieną XMI formatu išsaugotą modelį į kitą.

Kita kalba, galinti transformuoti modelius yra QVT. Tai eksperimentinė kalba, pasiūlyta kaip vienas iš galimų standartų modelių transformacijai MDA architektūroje. Šiame darbe bus panagrinėtos šios kalbos galimybės transformuoti modelius, taip pat bus palyginta su XSLT kalba.

Kaip realūs modelių pavyzdžiai buvo pasirinktos agregatinės sistemos. Bus sukurtas agregatinės sistemos PIM modelis ir XSLT bei QVT transformavimo kalbų pagalba bus transformuojami į Java platformai specifinius modelius.

## 2. MDA ARCHITEKTŪRA

Pirmasis modelis, kurį MDA apibrėžia yra aukšto abstrakcijos lygio modelis kuris yra nepriklausomas nuo programinės įrangos realizavimo technologijos. Jis vadinamas PIM modeliu (*Platform Independent Model*). PIM modelis aprašo sistemos funkcionalumą. Šiame modelyje svarbu, kokią paskirtį atliks sistema, kaip ji veiks, ir visiškai nesvarbu, kaip ji bus realizuota.

Gautas PIM modelis yra transformuojamas į vieną ar daugiau PSM modelių (*Platform Specific Models*). PSM modelis yra skirtas aprašyti sistemą, kaip ji bus realizuota naudojant konkrečią technologiją. PSM modelis yra generuojamas kiekvienai technologijai, kuria sistema bus realizuota. Kadangi šiuolaikinės sistemos apima daugiau negu vieną technologiją, tai dažniausiai naudojamas atvejis bus vienas PIM modelis ir keli PSM modeliai.

Gautas PSM modelis yra transformuojamas į programos kodą. Kadangi PSM modelis glaudžiai atitinka savo technologija, tai ši transformacija yra gana nesudėtinga.

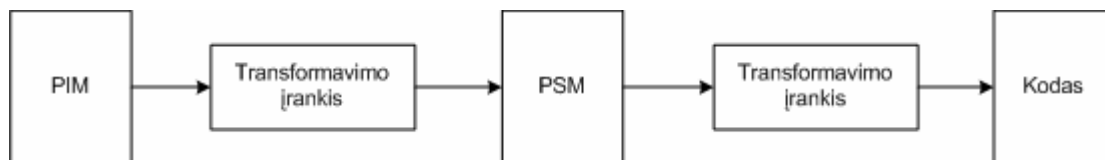
MDA apibrėžia PIM, PSM modelius ir programos kodą ir nustato, kaip jie yra susieti vienas su kitu. Pirmiausia kuriamas PIM modelis, tada jis transformuojamas į PSM modelį, kuris transformuojamas į programos kodą. Sudėtingiausias veiksmas MDA procese yra PIM modelio transformavimas į PSM modelius.

PIM, PSM modeliai ir programos kodas yra parodomi kaip rezultatai skirtinguose programavimo ciklo žingsniuose. Svarbiausia tai kad jie atspindi skirtingus abstrakcijos lygius sistemos specifikacijoje. Galimybė transformuoti aukšto lygio PIM modelį į PSM modelį padidina abstrakcijos lygį kuriame programuotojas gali dirbti. Tai leidžia programuotojui dirbti su sudėtingesnėmis sistemomis įdedant mažiau pastangų.

Gali atrodyti kad MDA procesas yra labai panašus į tradicinį programavimą. Tačiau čia yra vienas esminis skirtumas. Tradiciškai transformacijos iš vieno modelio į kitą modelį arba iš modelio į kodą yra daugiausia atliekamos rankomis. Daug priemonių gali generuoti tam tikrą kodą iš modelio, tačiau dažniausiai sugeneruoja tik programos šabloną, kur didžiąją dalį kodo tenka vis tiek parašyti patiems.

Priešingai, MDA transformacijos visada yra atliekamos automatizuotai. Daug yra programų galinčių transformuoti PSM modelį į kodą. Šiame žingsnyje MDA neatneša nieko naujo. Žinant

kad PSM modelis yra glaudžiai susietas su kodu, tokios transformacijos yra gana paprastos. Nauja MDA architektūroje yra tai kad transformacija iš PIM į PSM kodą yra automatizuota taip pat. Štai čia ir yra didžiausias MDA pranašumas.



### 3. pav. Pagrindiniai etapai MDA programinės įrangos kūrimo procese

MDA architektūroje pagrindinis programuotojo tikslas yra PIM modelio sudarymas. PSM modeliai yra sugeneruojami transformavimo įrankių pagalba iš PIM modelio. Žinoma, kažkam vis tiek teks parašyti transformaciją, o tai yra sudėtinga užduotis. Tačiau tokią transformaciją reikės parašyti tik vieną kartą ir ji galės būti pritaikoma kuriant daugelį sistemų.

Dauguma programuotojų didžiausią dėmesį skirs PIM modelio kūrimui. Kadangi jie tai gali nepriklausomai nuo realizavimo platformos detalių ir specifikos, atsiranda daug techninių detalių su kuriomis jiems neteks susidurti. Šios techninės detalės automatiškai bus įdėtos PIM modelio transformacijos į PSM modelį metu. Tai padidina produktyvumą dviem būdais.

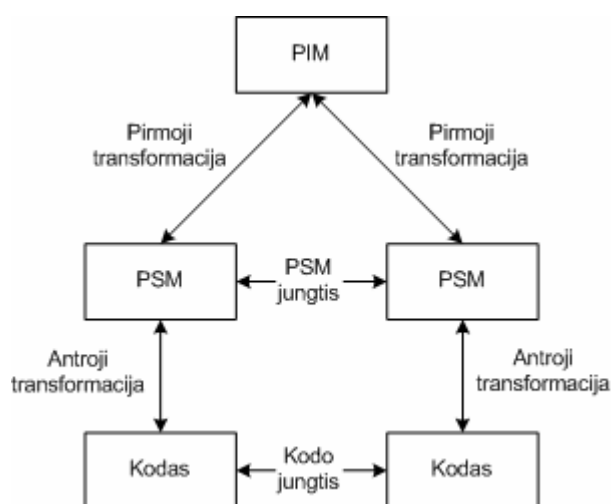
Pirmiausia PIM modelio kūrėjai turės mažiau darbo, nes nereikės projektuoti ir aprašyti platformos specifinių detalių, jos jau yra aprašytos transformacijos aprašyme. Daug mažiau rašymo bus PSM modelio ir programos kodo lygyje, nes didžioji dalis kodo bus jau sugeneruota iš PIM modelio.

Sekantis produktyvumo pagerėjimas yra tai kad programuotojai galės pagrindinį dėmesį skirti ne programos kodui bet PIM modeliui, tai reiškia galės skirti daugiau dėmesio verslo problemų sprendimui. Tai leis suskurti sistemas kurios geriau atitinka galutinių vartotojų reikalavimus. Galutiniai vartotojai gaus geresnį funkcionalumą per trumpesnę laiką.

Toks produktyvumo lygis gali būti pasiektas tik naudojant priemones kurios pilnai automatizuoja PSM modelio generavimą iš PIM modelio. Tai reiškia kad didžioji dalis informacijos apie sistemą turi būti PIM modelyje arba transformacijos aprašyme. Kadangi aukšto lygio modelis nebėra „tik popierius“, bet tiesiogiai susietas su generuojamu kodu, PIM modelio reikalavimai užbaigtumui ir teisingumui yra didesni negu tradiciniame programavime.

Portatyvumas (*portability*) MDA architektūroje yra pasiekiamas kuriant nuo platformos nepriklausomus PIM modelius. Tas pats PIM modelis gali būti transformuojamas į daugelį PSM modelių skirtingoms platformoms. Koks portatyvumo lygis bus pasiektas priklausys nuo automatizuotų transformavimo priemonių kurios yra prieinamos. Neabejotina, kad populiarioms platformoms bus parašyta daug priemonių, tačiau mažai populiarioms platformoms gali tekti aprašinėti transformacijas patiems. Atsiradus naujoms technologijoms ir platformoms užteks parašyti transformacijas. Taip bus galima greitai įdiegti sistemas su naujomis technologijomis naudojant senus jau egzistuojančius PIM modelius.

Kai PSM modeliai yra generuojami skirtingoms platformoms, jie negali tiesiogiai bendrauti vienas su kitu. Vienu ar kitu būdu, reikia transformuoti vienos platformos koncepciją į koncepciją naudojamą kitoje platformoje. MDA sprendžia šią problemą generuodama ne tik PSM modelius bet ir ryšius tarp tų modelių.



#### 4. pav. MDA sistemų tarpusavio veikimas naudojant jungiamuosius ryšius

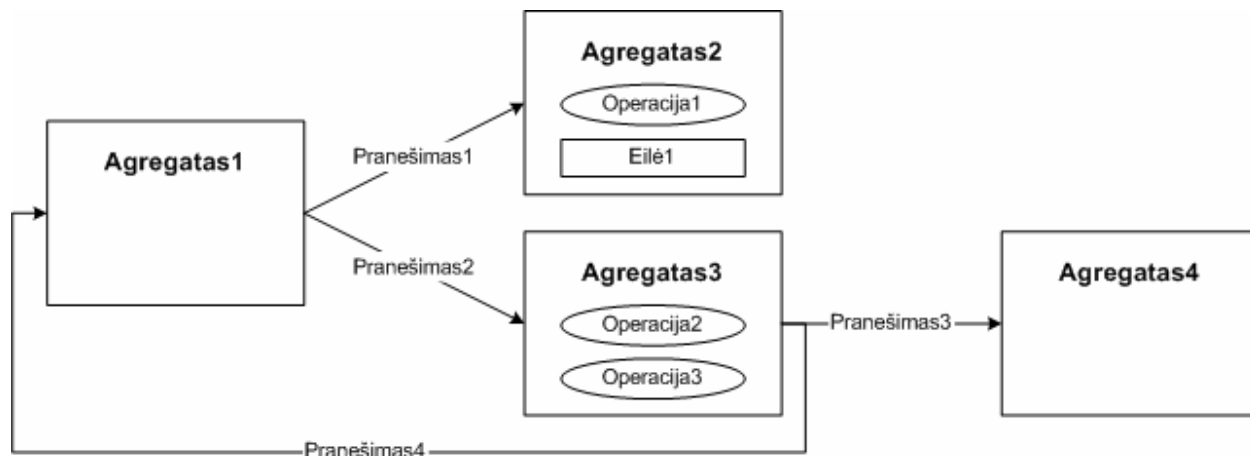
Naudojant MDA programinės įrangos kūrimo procesą, programuotojai gali dėmesį sutelkti į PIM modelį, kuris yra aukštesniam abstrakcijos lygyje negu programos kodas. PIM modelis yra naudojamas sugeneruoti PSM modelį, iš kurio generuojamas kodas. Kadangi PIM modelis tiesiogiai atspindi kodą, tai jis atitinka aukšto lygio dokumentaciją, kuri yra reikalinga bet kuriai programinės įrangos sistemai.

Didelis skirtumas yra tai kad PIM modelis nėra apleidžiamas baigus rašyti. Pakeitimai daromi sistemoje, bus atlikti keičiant PIM modelį ir regeneruojant PSM modelį ir kodą. Jei pakeitimai

bus atliekami PSM modelyje, atitinkamai bus pakeistas ir PIM modelis. Todėl aukšto lygio dokumentacija visada atitiks programos kodą.

### 3. AGREGATINĖS SISTEMOS

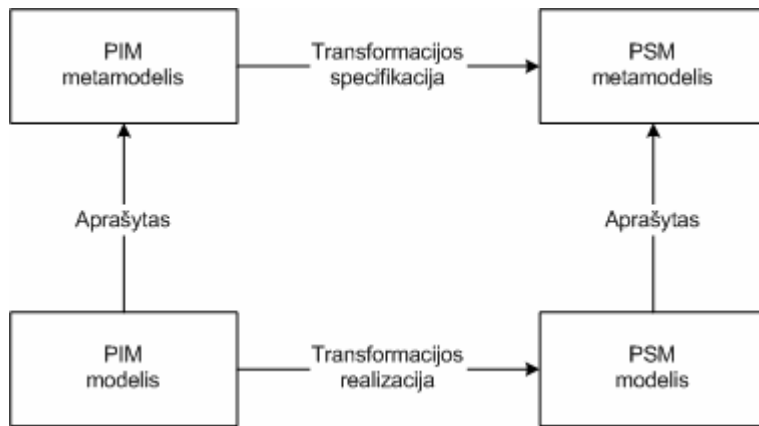
Žemiau esančiame paveiksle pavaizduotas imitacinis agregatinės sistemos modelis. Sistemą sudaro keli tarpusavyje sujungti agregatai, kurie bendrauja tarpusavyje siusdami pranešimus ir atlieka tam tikras operacijas.



5. pav. Agregatinės sistemos modelis

Šis agregatinės sistemos modelis bus naudojamas nagrinėjant modelių transformacijas MDA architektūroje. Bus sudarytas šios sistemos nuo realizacijos sistemos nepriklausantis PIM modelis, kuris naudojant XSLT ir QVT transformacijas bus transformuotas į sistemai specifinius PSM modelius.

Norint sudaryti PIM modelį, turime apibrėžti iš kokių elementų jį sudarysime. Tam sudaromas PIM metamodelis. PIM metamodelis apibrėžia iš kokių elementų bus sudarytas PIM modelis. Sudarant PSM modelį, taip pat bus sudarytas PSM metamodelis. Naudojantis PIM ir PSM metamodeliais bus sudarytos modelių transformavimo taisyklės. Jos nurodys koks PIM metamodelio elementas turi būti transformuojamas į kokį PSM metamodelio elementą. Žemiau esančiame paveiksle parodyta kaip susiję PIM ir PSM metamodeliai.



6. pav. PIM ir PSM metamodeliai

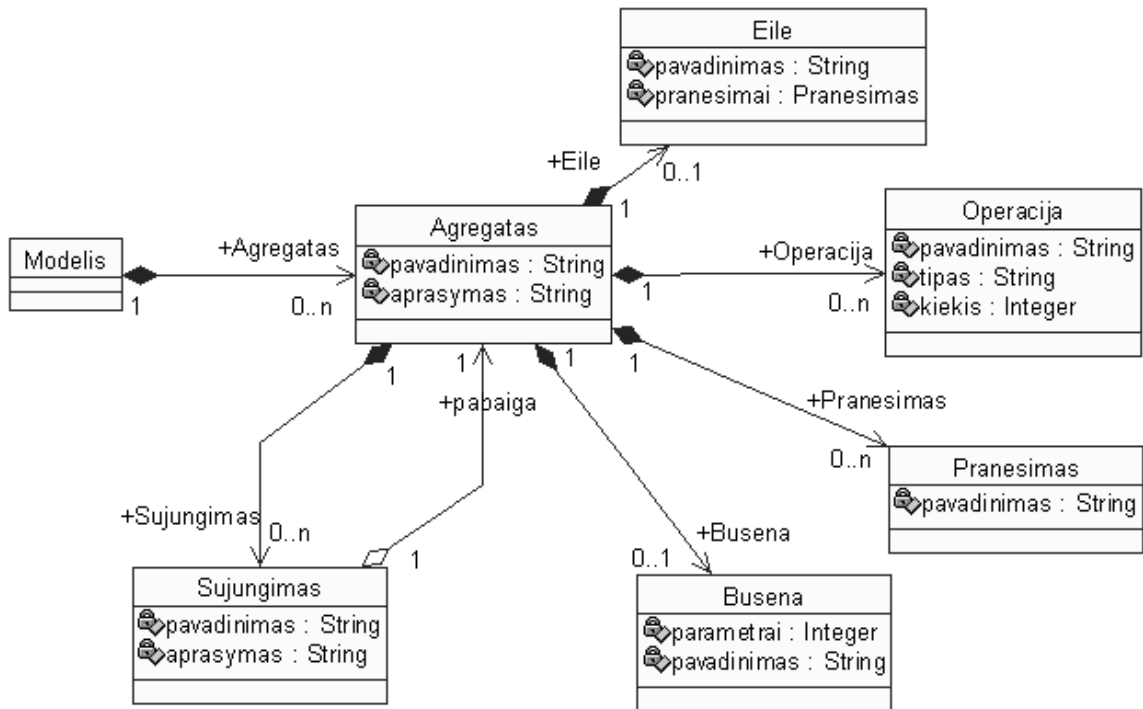
Agregatinės sistemos elementai yra:

- Agregatas
- Pranešimas
- Operacija
- Sujungimas
- Eile
- Busena

Agregatai gali turėti operacijas ir sujungimus. Sujungimai turi nuorodas į agregatus su kuriais yra susieti. Sujungimai taip pat gali turėti pranešimus, kurie yra siunčiami tarp susietų agregatų.

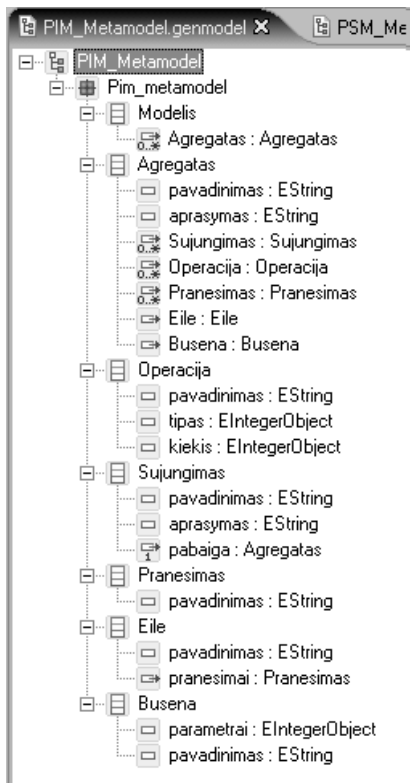
Žemiau parodytas agregatinės sistemos PIM metamodelis.





7. pav. PIM metamodelis

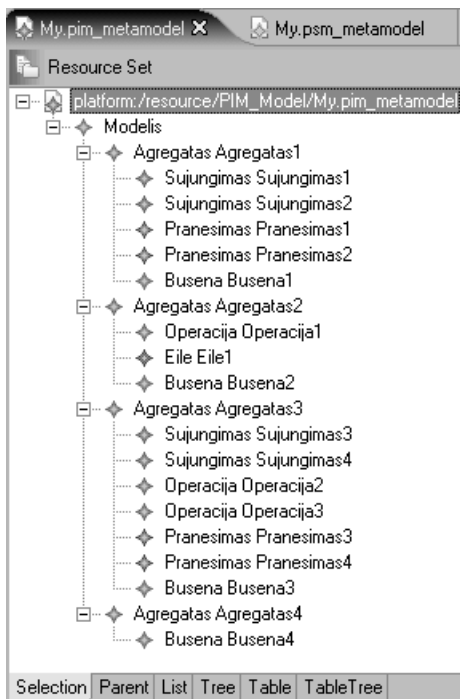
Iš UML diagramos sugeneruojame PIM metamodelio redaktorių. Tam naudojame Eclipse platforma ir EMF (*Eclipse Modeling Framework*) įrankius. PIM metamodelio redaktorių naudosime metamodelio išsaugojimui XMI formatu ir PIM modelio sugeneravimui. PIM metamodelio redaktorių Eclipse aplinkoje pavaizduotas žemiau esančiame paveiksle.



8. pav. PIM metamodelis Eclipse aplinkoje

PIM metamodelio aprašas XMI formatu pateiktas prieduose.

Naudojantis šiuo metamodeliu sudaromas PIM modelis.



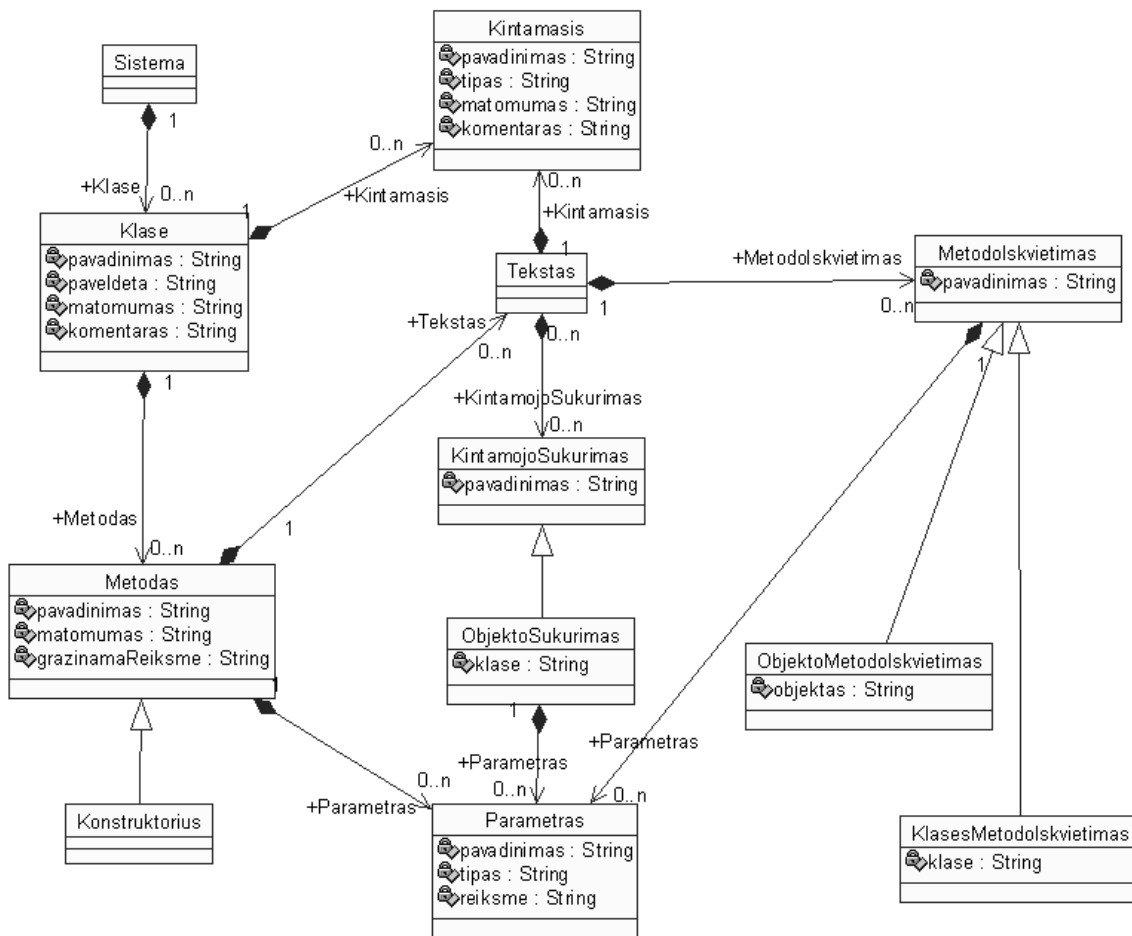
9. pav. PIM modelis Eclipse aplinkoje

Agregatinės sistemos modelis bus transformuojamas į Java platformą. Tam sudaromas PSM metamodelis aprašantis Java kalbos elementus. Į PSM modelį įtraukti tik tie elementai, kurie būtini šioje agregatinės sistemos transformacijoje.

Sudarant PSM modelį buvo nustatyti tokios modelio elementų savybės:

- Visa sistema yra sudaryta iš klasių.
- Klasė gali turėti metodus ir kintamuosius.
- Klasė gali būti paveldėta iš kitos klasės.
- Klasės konstruktorius išskiriamas kaip atskiras metodas.
- Metodas gali turėti parametrus ir tekstą.
- Tekste gali būti kuriami paprasti kintamieji, klasių objektai ir kviečiame metodai.
- Klasės, metodai, kintamieji gali turėti matomumo parametrus (public, protected, private)

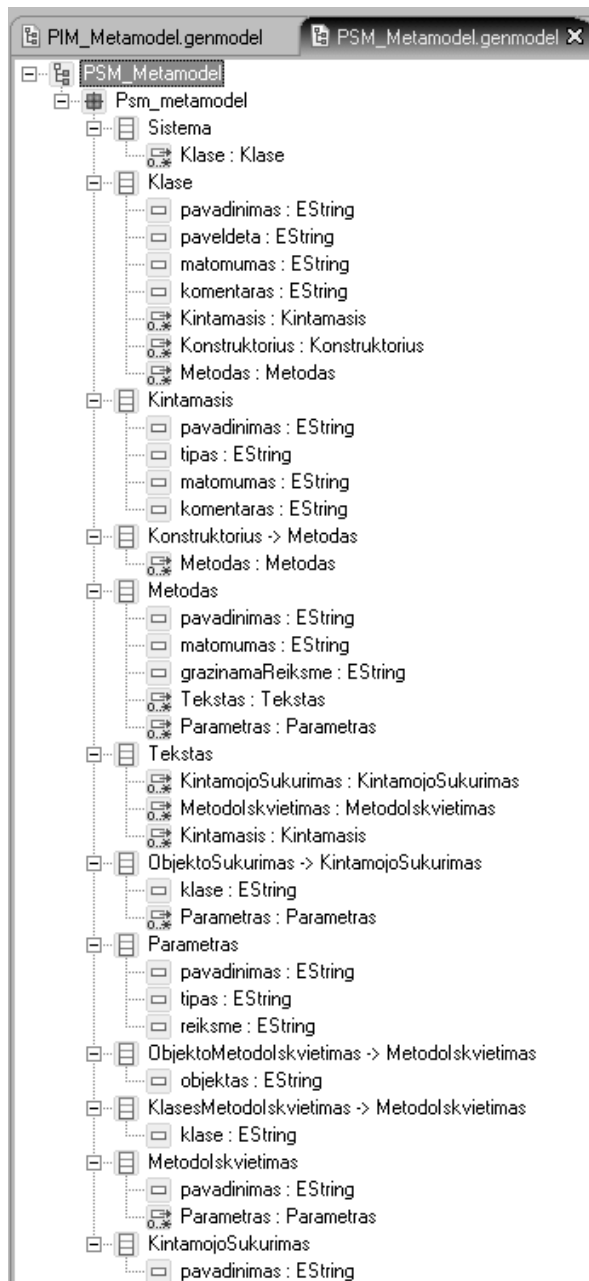
PSM modelio metamodelis parodytas žemiau esančiame paveiksle.



10. pav. PSM metamodelis

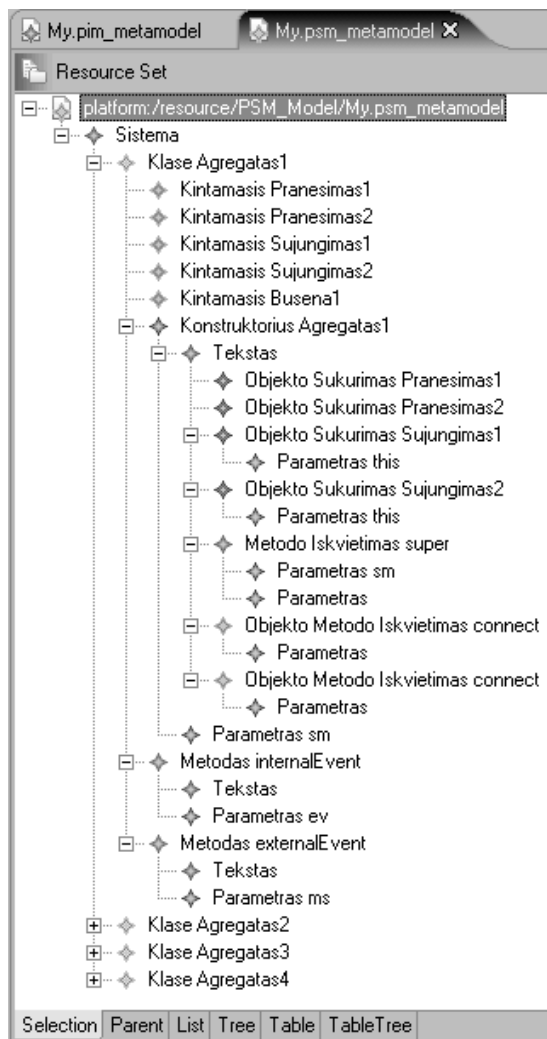
PSM metamodelio elementų aprašymas:

Iš UML diagramos sugeneruojame PSM metamodelio redaktorių. Tam naudojame Eclipse platforma ir EMF (*Eclipse Modeling Framework*) įrankius. PSM metamodelio redaktorių naudosime metamodelio išsaugojimui XMI formatu ir PSM modelio sugeneravimui (kad patikrinti transformacijos veikimo teisingumą). PSM metamodelio redaktorių Eclipse aplinkoje pavaizduotas žemiau esančiame paveiksle.

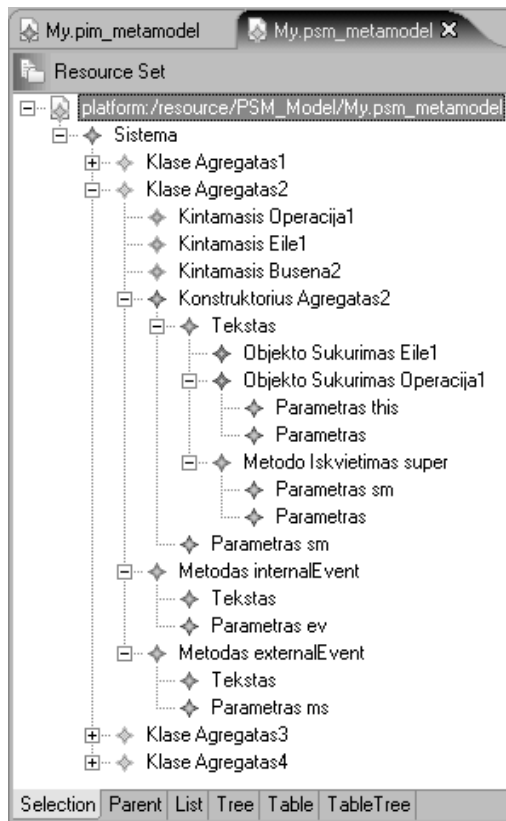


**11. pav. PSM metamodelis Eclipse aplinkoje**

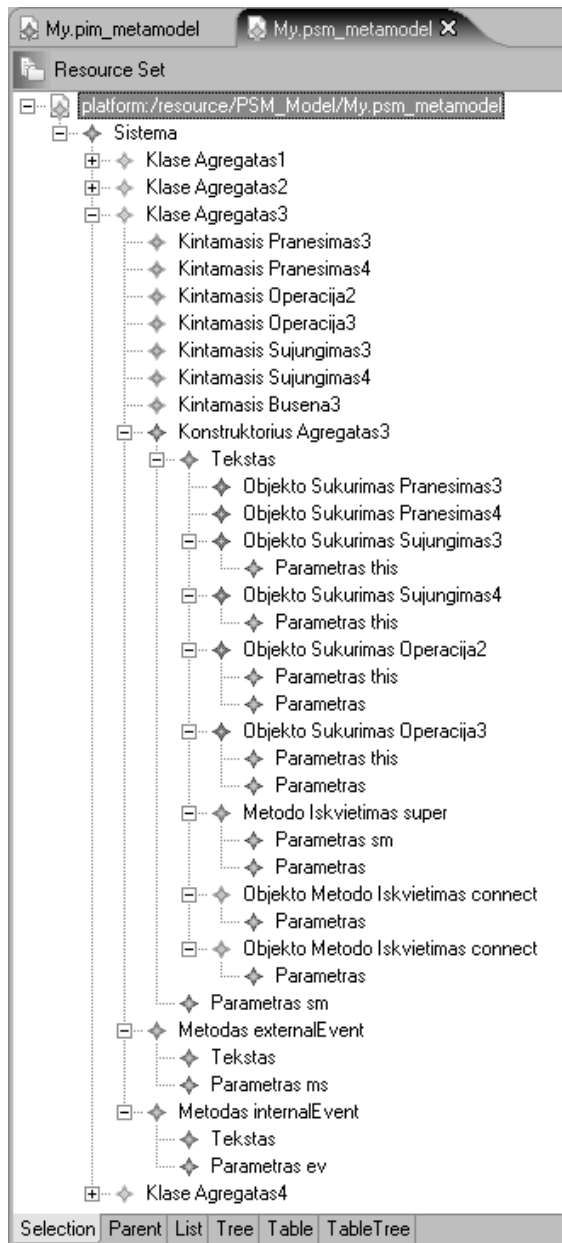
PSM metamodelio aprašas XMI formatu pateiktas prieduose. Naudojantis šiuo metamodelio redaktoriumi sumodeliuojamas PSM modelis. Kadangi PSM modelis yra labai didelis, jis pateikiamas dalimis, parodant kiekvieno agregato po vieną klasę.



12. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas1 klasė.

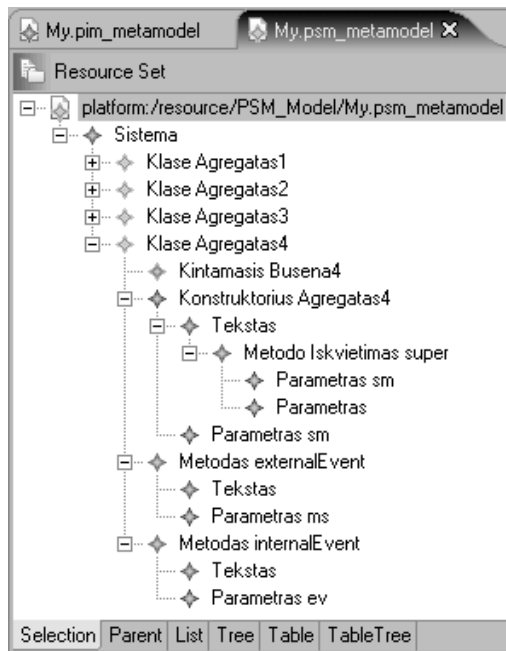


13. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas2 klasė.



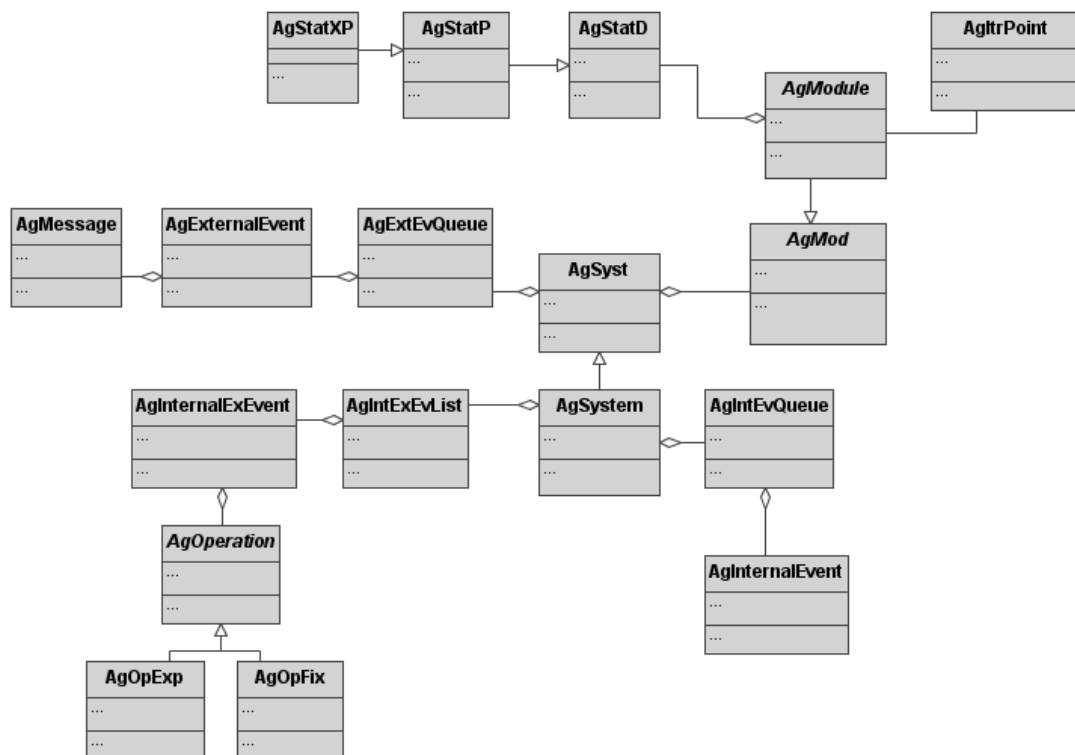
14. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas3 klasė.





15. pav. PSM modelis Eclipse aplinkoje. Parodyta Agregatas4 klasė.

Šis PSM modelis sudarytas taip, kad galėtų naudoti jau egzistuojančią agregatinių sistemų modeliavimo biblioteką. Java kodo atitinkančio PSM modelio klasę Agregatas1 pavyzdys parodytas prieduose. Bibliotekos klasių diagrama pateikta žemiau esančiame paveiksle.



16. pav. Modeliavimo biblioteka.

Paveiksle pateikta visa modeliavimo biblioteka. Bibliotekoje yra jau realizuota:

- Agregatų klasės
- Sistemos modelis
- Pranešimai
- Įvykiai
- Eilės
- Operacijos
- Statistikos rinkimas

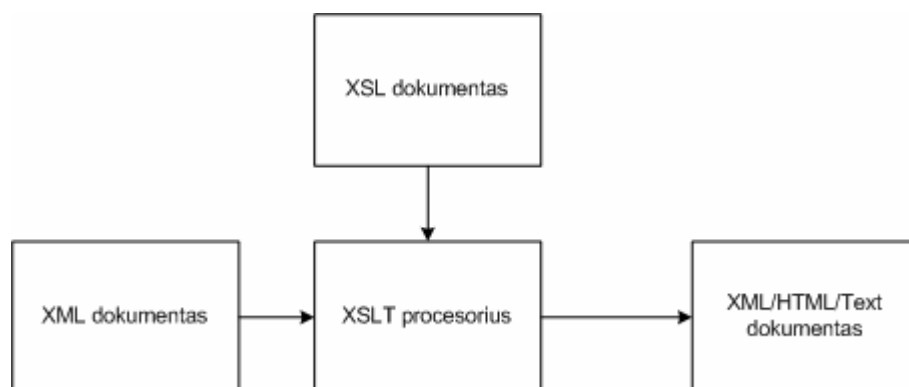
Agregatai realizuoti klasėje AgModule. Sistemos modelis – AgSystem. Pranešimai – AgMessage. Įvykiai – AgInternalExEvent. Operacijos – AgOperation, AgOpExp, AgOpFix. Statistikos surinkimas – AgStatD, AgStatX, AgStatXP. Taip pat realizuotos atitinkamų įvykiu eilės.

## 4. XSLT

### 4.1. XSLT transformacijos

XSLT (*Extensible Stylesheet Language Transformations*) yra kalba skirta transformuoti XML dokumentus. XSLT yra sukurta naudojimui kaip XSL (*Extensible Stylesheet Language*) kalbos dalis.

Žemiau esančiame paveiksle parodytas bendras XSLT naudojimo atvejis.



17. pav. Bendras XSLT naudojimo atvejis

XSLT procesoriui yra paduodamas XML dokumentas kuris bus transformuojamas ir XSL dokumentas, kuris nurodo kaip transformuotas dokumentas atrodys. Rezultate gaunamas naujas dokumentas ( dažniausiai XML, HTML ar tekstinio tipo dokumentas).

XSLT transformacijos taip pat gali veikti nepriklausomai nuo XSL.

### 4.2. Modelių transformacijos

Gautas PIM modelis išsaugomas XML formatu ir transformuojamas į PSM modelį naudojant XSLT transformaciją. Žemiau pateiktas kiekvieno PIM modelio elemento transformavimo aprašymas.

#### 4.2.1. Elemento Operacija transformacija

PIM metamodelio elementas Operacija turi šiuos elementus:

- Pavadinimas
- Tipas
- Kiekis

Žemiau esančioje lentelėje parodyta į kokius PSM metamodelio elementus šie elementai yra transformuojami.

**1 lent. Elemento Operacija transformacija**

PIM Metamodelio elementas	PSM Metamodelio elementas	Transformavimo paskirtis
Operacija	Kintamasis	Aprašomas klasės kintamasis
Operacija.pavadinimas	Kintamasis.pavadinimas	Kintamojo pavadinimas
Operacija.tipas	Kintamasis.tipas	Kintamojo tipas
Operacija	ObjektoSukurimas	Klasės konstruktoriuje sukuriamas elementą Operacija atitinkantis objektas.
Operacija.tipas	ObjektoSukurimas.klase	Kuriamo objekto klasė
Operacija.kiekis	ObjektoSukurimas.parametras	Konstruktoriaus parametras

Tekstinis šios XSLT transformacijos aprašas pateiktas prieduose.

#### 4.2.2. Elemento Sujungimas transformacija

PIM metamodelio elementas Sujungimas turi šiuos elementus:

- Pavadinimas
- Aprašymas
- Pabaiga

Žemiau esančioje lentelėje parodyta į kokius PSM metamodelio elementus šie elementai yra transformuojami.

**2 lent. Elemento Sujungimas transformacija**

PIM Metamodelio elementas	PSM Metamodelio elementas	Transformavimo paskirtis
Sujungimas	Kintamasis	Aprašomas klasės kintamasis
Sujungimas.pavadinimas	Kintamasis.pavadinimas	Kintamojo pavadinimas
Sujungimas.aprašymas	Kintamasis.komentaras	Kintamąjį paaiškinantis komentaras
Sujungimas	ObjektoSukurimas	Klasės konstruktoriuje sukuriamas elementą Sujungimas atitinkantis objektas.
Sujungimas	ObjektoMetodoIskvietimas	Iškviečiamas objekto metodas
Sujungimas.pabaiga	ObjektoMetodoIskvietimas.parametras	Metodo parametras

Tekstinis šios XSLT transformacijos aprašas pateiktas prieduose.

#### 4.2.3. Elemento Pranesimas transformacija

PIM metamodelio elementas Pranesimas turi šiuos elementus:

- Pavadinimas

Žemiau esančioje lentelėje parodyta į kokius PSM metamodelio elementus šie elementai yra transformuojami.

**3 lent. Elemento Sujungimas transformacija**

<b>PIM Metamodelio elementas</b>	<b>PSM Metamodelio elementas</b>	<b>Transformavimo paskirtis</b>
Pranesimas	Kintamasis	Aprašomas klasės kintamasis
Pranesimas.pavadinimas	Kintamasis.pavadinimas	Kintamojo pavadinimas
Pranesimas	ObjektoSukurimas	Klasės konstruktoriuje sukuriama elementą Pranesimas atitinkantis objektas.

Tekstinis šios XSLT transformacijos aprašas pateiktas prieduose.

#### 4.2.4. Elemento Eile transformacija

PIM metamodelio elementas Eile turi šiuos elementus:

- Pavadinimas
- Pranesimai

Žemiau esančioje lentelėje parodyta į kokius PSM metamodelio elementus šie elementai yra transformuojami.

**4 lent. Elemento Sujungimas transformacija**

<b>PIM Metamodelio elementas</b>	<b>PSM Metamodelio elementas</b>	<b>Transformavimo paskirtis</b>
Eile	Kintamasis	Aprašomas klasės kintamasis
Eile.pavadinimas	Kintamasis.pavadinimas	Kintamojo pavadinimas
Eile.pranesimai	Tai vidinis elementą Eile atitinkančios klasės AgQueue kintamasis. Jis jau realizuotas šioje klasėje, todėl nėra transformuojamas.	Šiame kintamajame saugomi eilėje esantys pranešimai.

Tekstinis šios XSLT transformacijos aprašas pateiktas prieduose.

#### 4.2.5. Elemento Busena transformacija

PIM metamodelio elementas Busena turi šiuos elementus:

- Pavadinimas
- Parametrai

Žemiau esančioje lentelėje parodyta į kokius PSM metamodelio elementus šie elementai yra transformuojami.

5 lent. Elemento Sujungimas transformacija

PIM Metamodelio elementas	PSM Metamodelio elementas	Transformavimo paskirtis
Busena	Kintamasis	Aprašomas klasės kintamasis
Busena.pavadinimas	Kintamasis.pavadinimas	Kintamojo pavadinimas
Busena.parametrai	Būseną nusakantys parametrai. Tai jau realizuota modeliavimo bibliotekoje, todėl nėra transformuojama.	Būseną nusakantys parametrai

Tekstinis šios XSLT transformacijos aprašas pateiktas prieduose.

#### 4.2.6. Elemento Agregatas transformacija

PIM metamodelio elementas Agregatas turi šiuos elementus:

- Pavadinimas
- Aprašymas
- Sujungimas
- Operacija
- Pranesimas
- Eile
- Busena

Žemiau esančioje lentelėje parodyta į kokius PSM metamodelio elementus šie elementai yra transformuojami.

6 lent. Elemento Sujungimas transformacija

PIM Metamodelio elementas	PSM Metamodelio elementas	Transformavimo paskirtis
Agregatas	Klasė	Sukuriamą agregatą modeliuojanti klasė.
Agregatas.pavadinimas	Klasė.pavadinimas	Klasės pavadinimas
Agregatas.aprašymas	Klasė.komentaras	Klasę paaiškinantis komentaras
Agregatas.Sujungimas	Transformuojama pagal aukščiau aprašytas taisykles.	Elementą Sujungimas atitinkantis objektas kuriame saugoma informacija apie agregatus su kuriais šis agregatas yra sujungtas.
Agregatas.Operacija	Transformuojama pagal aukščiau aprašytas taisykles.	Elementą Operacija atitinkantis objektas kuriame

		saugoma informacija apie agregato atliekamas operacijas.
Agregatas.Pranesimas	Transformuojama pagal aukščiau aprašytas taisykles.	Elementą Pranesimas atitinkantis objektas kuriame saugoma informacija apie agregato siunčiamus pranešimus.
Agregatas.Eile	Transformuojama pagal aukščiau aprašytas taisykles.	Elementą Eile atitinkantis objektas kuriame saugoma informacija apie agregato dar neapdorotus pranešimus.
Agregatas.Busena	Transformuojama pagal aukščiau aprašytas taisykles.	Elementą Busena atitinkantis objektas kuriame saugoma informacija apie agregato būseną.

Tekstinis šios XSLT transformacijos aprašas pateiktas prieduose.

### 4.3. XSLT transformacijos apibendrinimas

XSLT transformacijos aprašas yra sudėtingas ir sunkiai skaitomas. Šią transformaciją sunku aprašyti rankomis, todėl dideliems modeliams labiau pritaikoma tik generuojant automatiškai. Transformacijos aprašas sunkiai skaidomas į atskirus blokus, todėl sudėtinga ją pakartotinai panaudoti aprašant kitas transformacijas. XSLT privalumas yra tai kad yra sukurta daug priemonių palengvinančių šių transformacijų kūrimą.

## 5. QVT

### 5.1. QVT standartas

QVT (*Queries/Views/Transformations*) yra OMG standartas skirtas MDA modelių transformacijai. Šiuo metu QVT dar nėra patvirtintas, todėl bus nagrinėjamas vienas iš galimų standarto variantų, pasiūlytas QVT-Partners grupės.

Užklausos (*queries*), vaizdai (*views*) ir transformacijos yra subjektai kurie bus labai svarbūs MDA architektūros sėkmei. Šie trys QVT standarto pasiūlymo aspektai apibrėžia platesnę problemą: galimybė manipuliuoti modeliais yra būtina kad užtikrinti pilną MDA architektūros realizaciją. Šis QVT standarto pasiūlymas siekia standartinio modelių valdymo sprendimo.

QVT subjektų aprašymas:

- Užklausos iš duoto modelio išrenka reikalingus elementus.
- Vaizdai yra modeliai kurie yra gaunami iš kitų modelių.
- Transformacijos atnaujina ta pati modelį arba iš jo generuoja kitą modelį.

Svarbu pažymėti kad užklausos, vaizdai ir transformacijos gali būti suskirstyti į dvi skirtingas grupes. Užklausos ir transformacijos atlieka veiksmus su modeliais rezultate gaunant naują arba pakeistą modelį. Priešingai, vaizdai patys yra modeliai ir yra susieti su modeliais iš kurių jie yra sukurti. Užklausos ir transformacijos gali kurti vaizdus, bet patys vaizdai yra neatlieka jokių veiksmų.

Užklausų aprašymui siūloma naudoti praplėstą OCL 2.0 kalbos versiją. OCL kalbos naudojimas turi keletą privalumų: vartotojai jau yra susipažinę su šia kalba, todėl nereikės mokytis naujos kalbos, taip pat yra sukurta daug priemonių palaikančių OCL kalbą.

### 5.2. Transformacijos

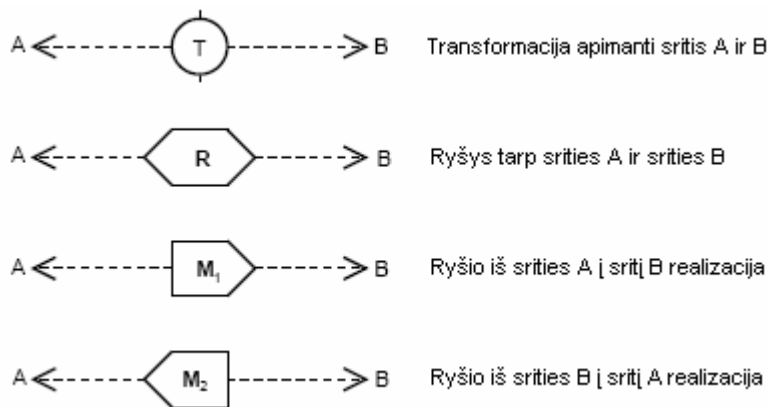
Išskiriami du skirtingi transformacijų tipai: ryšiai (*relations*) ir ryšių realizacijos (*mappings*).



Ryšiai yra daugiakrypčių transformacijų specifikacijos. Ryšiai nėra vykdomi, jie negali sukurti ar pakeisti modelio, jie skirti tam kad patikrinti ar vienas modelis atitinka kitą. Dažniausiai ryšiai yra naudojami sistemos specifikavimo metu, arba ryšių realizacijos teisingumui tikrinti.

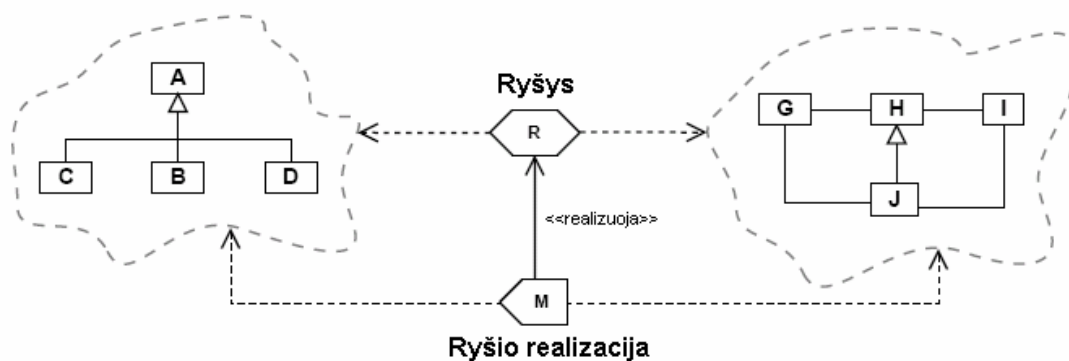
Ryšių realizacijos yra transformacijų realizacijos. Jos yra vienakryptės ir gali gražinti rezultatus. Ryšių realizacijos gali realizuoti bet koki ryšių kieki. Tuo atveju jos turi atitikti ryšius kuriuos realizuoja.

Žemiau esančiame paveiksle parodyti žymėjimai naudojami QVT standarto pasiūlyme.



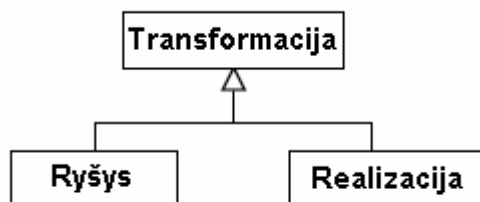
18. pav. Žymėjimai naudojami QVT standarto pasiūlyme

Žemiau esančiame paveiksle parodytas ryšys R siejantis dvi sritis (*domains*). Taip pat yra pavaizduota ir ryšio R realizacija M. Kadangi M turi kryptį, ji transformuoja modelio elementus iš dešinėje esančios srities į kairėje esančią sritį.



19. pav. Aukšto lygio ryšis ir jo realizacija

Žemiau esančiame paveiksle parodyta kaip transformacijos, ryšiai ir jų realizacijos susijusios tarpusavyje. Kadangi ryšys ir jo realizacija yra paveldėta iš transformacijos, kai yra šnekama apie transformaciją, turima galvoje arba ryšį arba jo realizaciją, nesvarbu kuri.



20. pav. Transformacijos, ryšiai ir jų realizacijos

Transformacijų aprašymo ir jų realizacijų atskyrimas turi keletą privalumų. Tai leidžia modelių kūrėjui:

- naudoti ryšius pradinėje sistemos kūrimo fazėje kai realizavimo detalės dar nėra pilnai žinomos.
- parašyti daug to pačio ryšio realizacijų. Realizacijos gali būti rašomos net naudojant skirtingas kalbas.
- Kurti ryšius iš jau egzistuojančių realizacijų.

Ryšiams ir jų realizacijoms aprašyti siūloma nauja kalba MTL (*Model Transformation Language*). MTL naudoja šablonų sulyginimo (*pattern matching*) būdą kaip vieną iš pagrindinių veiksnių leidžiančių sukurti sudėtingas transformacijas. Tos pačios kalbos naudojimas ryšių ir jų realizacijų aprašymui turi daug privalumų, iš kurių vienas svarbiausių yra tai kad greičiau galima išmokti naudoti transformacijas. MTL kalba turi tiek grafinę tiek tekstinę formą. Grafinė forma labiausiai naudinga kai naudojama kaip tekstinės formos abstrakcija.

Šablonų sulyginimo būdas jau yra naudojamas tokiose transformacijų sistemose kaip XSLT ar tekstinės reguliarios išraiškos (*regular expressions*). Pagrindinė šablonų sulyginimo idėja yra leisti paduoti trumpas išraiškas kaip įėjimo duomenis. Duomenys kurie atitinka šabloną yra išrenkami ir gražinami kaip rezultatas.

Šablonų sulyginimo būdas iš prigimties yra kompromisas tarp išraiškiningumo (*expressivity*) ir glaustumo. Taip pat kaip ir dauguma šablonų naudojančių kalbų, ši kalba nėra visada pats geriausias būdas išreikšti tam tikros transformacijos aspektus. Tuo tikslu transformacijos

aprašymas yra sudaromas iš šablonų ir sąlygų prie kurių jie yra tikrinami. Taip pat yra įvedamas ir bendros sąlygos tikrinimas kuris apima visą sritį.

Bendra ryšio aprašymo forma:

```
relation R
{
    domain { pattern1 when condition1 }
    domain { patternn when conditionn }
    when { condition }
}
```

#### 21. pav. Ryšio aprašymo forma

Šis ryšys susieja dviejų modelių elementus. Galutinė sąlyga yra globali sąlyga kuri apima visas sritis (*domains*). Ryšys yra tenkinamas tik tada kai visos sritys yra tenkinamos.

Bendra ryšio realizacijos aprašymo forma:

```
mapping M refines R
{
    domain { pattern1 when condition1 }
    body {...}
}
```

#### 22. pav. Ryšio realizacijos aprašymo forma

Kadangi ryšio realizacija yra vienkryptė (ji transformuoja vieno modelio elementus į kitą modelį), tai ryšio realizacijos aprašyme išlieka tik vienas modelis iš kurio imami elementai. Kito modelio elementai formuojami *body* lauke.

### 5.3. Modelių transformacijos

Aprašant modelių transformacijas, pirmiausia bus nustatomi ryšiai tarp metamodelių elementų. Bus apibrėžti šie ryšiai:

- PIM metamodelio elemento Operacija ryšys su PSM metamodelio elementais.
- PIM metamodelio elemento Sujungimas ryšys su PSM metamodelio elementais.
- PIM metamodelio elemento Pranesimas ryšys su PSM metamodelio elementais.

- PIM metamodelio elemento Eile ryšys su PSM metamodelio elementais.
- PIM metamodelio elemento Busena ryšys su PSM metamodelio elementais.

Nustačius ryšius, bus sudaromos jų realizacijos. Visiems ryšiai bus pavaizduoti tiek tekstinė, tiek grafinė forma.

Aprašius atskirų elementų ryšius, ir jų realizacijas, bus sudaryta pagrindinio PIM metamodelio elemento Agregatas transformacija į PSM metamodelio elementus.

### 5.3.1. Elemento Operacija transformacija

PIM metamodelio elementas Operacija turi šiuos elementus:

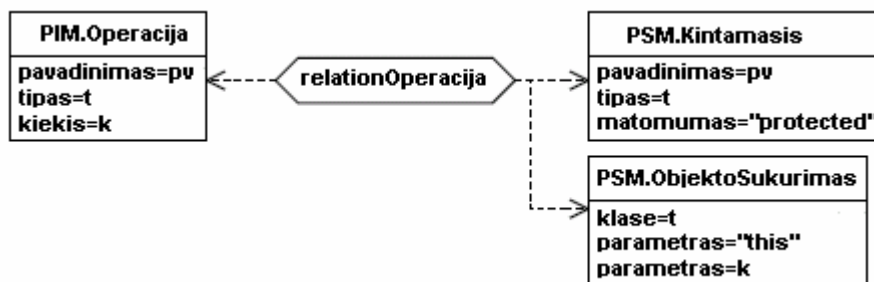
- Pavadinimas
- Tipas
- Kiekis

Šie elementai yra susiejami su PSM metamodelio elementais:

- Kintamasis
- ObjektoSukurimas

Tam yra sudaromas ryšys relationOperacija, kuris susieja šiuos metamodelių elementus.

Žemiau esančiuose paveiksluose pateikiama grafinė ir tekstinė transformacijos forma:



23. pav. Ryšio relationOperacija grafinė forma

```

relation relationOperacija{
  domain{(PIM.Operacija)[pavadinimas pv, tipas t, kiekis k]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = t, matomumas = "protected"]
    (PSM.ObjektoSukurimas)[klase = t, parametras = "this", parametras = k]
  }
}

```

24. pav. Ryšio relationOperacija tekstinė forma

```

mapping mappingBusena{
  domain{(PIM.Busena)[pavadinimas pv]}
  body{

```

```

(PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
}
}

```

25. pav. Ryšio realizacijos mappingOperacija tekstinė forma

### 5.3.2. Elemento Sujungimas transformacija

PIM metamodelio elementas Sujungimas turi šiuos elementus:

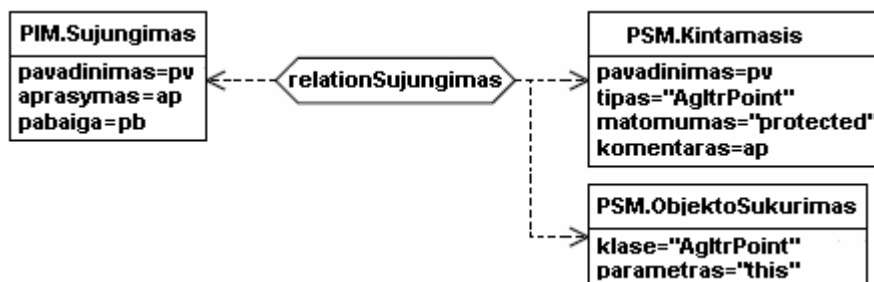
- Pavadinimas
- Aprasymas
- Pabaiga

Šie elementai yra susiejami su PSM metamodelio elementais:

- Kintamasis
- ObjektoSukurimas

Tam yra sudaromas ryšys relationSujungimas, kuris susieja šiuos metamodelių elementus.

Žemiau esančiuose paveiksluose pateikiama grafinė ir tekstinė transformacijos forma:



26. pav. Ryšio relationSujungimas grafinė forma

```

relation relationSujungimas{
  domain{(PIM.Sujungimas)[pavadinimas pv, aprasymas ap, pabaiga pb]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgltrPoint", matomumas = "protected",
komentaras = ap]
    (PSM.ObjektoSukurimas)[klase = "AgltrPoint", parametras = "this"]
  }
}

```

27. pav. Ryšio relationSujungimas tekstinė forma

```

mapping mappingBusena{
  domain{(PIM.Busena)[pavadinimas pv]}
  body{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
  }
}

```

28. pav. Ryšio realizacijos mappingSujungimas tekstinė forma

### 5.3.3. Elemento Pranesimas transformacija

PIM metamodelio elementas Pranesimas turi šiuos elementus:

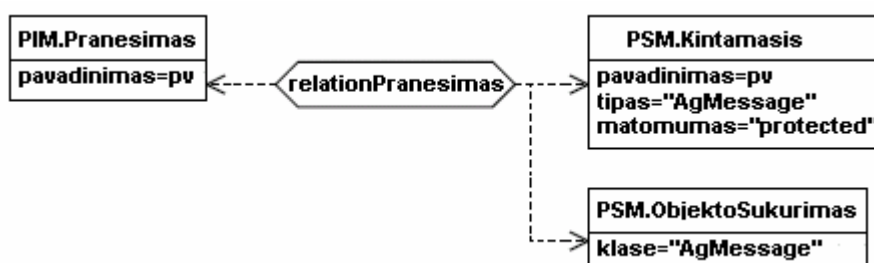
- Pavadinimas

Šie elementai yra susiejami su PSM metamodelio elementais:

- Kintamasis
- ObjektoSukurimas

Tam yra sudaromas ryšys relationPranesimas, kuris susieja šiuos metamodelių elementus.

Žemiau esančiuose paveiksluose pateikiama grafinė ir tekstinė transformacijos forma:



29. pav. Ryšio relationPranesimas grafinė forma

```
relation relationPranesimas{
  domain{ (PIM.Pranesimas) [pavadinimas pv] }
  domain{
    (PSM.Kintamasis) [pavadinimas = pv, tipas = "AgMessage", matomumas = "protected"]
    (PSM.ObjektoSukurimas) [klase = "AgMessage"]
  }
}
```

30. pav. Ryšio relationPranesimas tekstinė forma

```
mapping mappingBusena{
  domain{ (PIM.Busena) [pavadinimas pv] }
  body{
    (PSM.Kintamasis) [pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
  }
}
```

31. pav. Ryšio realizacijos mappingPranesimas tekstinė forma

### 5.3.4. Elemento Eile transformacija

PIM metamodelio elementas Eile turi šiuos elementus:

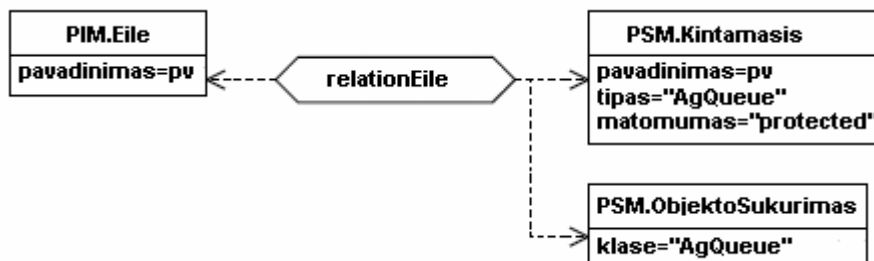
- Pavadinimas

Šie elementai yra susiejami su PSM metamodelio elementais:

- Kintamasis
- ObjektoSukurimas

Tam yra sudaromas ryšys relationEile, kuris susieja šiuos metamodelių elementus.

Žemiau esančiuose paveiksluose pateikiama grafinė ir tekstinė transformacijos forma:



32. pav. Ryšio relationEile grafinė forma

```

relation relationEile{
  domain{(PIM.Eile)[pavadinimas pv]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgQueue", matomumas = "protected"]
    (PSM.ObjektoSukurimas)[klase = "AgQueue"]
  }
}

```

33. pav. Ryšio relationEile tekstinė forma

```

mapping mappingBusena{
  domain{(PIM.Busena)[pavadinimas pv]}
  body{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
  }
}

```

34. pav. Ryšio realizacijos mappingEile tekstinė forma

### 5.3.5. Elemento Busena transformacija

PIM metamodelio elementas Busena turi šiuos elementus:

- Pavadinimas

Šie elementai yra susiejami su PSM metamodelio elementais:

- Kintamasis

Tam yra sudaromas ryšys relationBusena, kuris susieja šiuos metamodelių elementus.

Žemiau esančiuose paveiksluose pateikiama grafinė ir tekstinė transformacijos forma:



35. pav. Ryšio relationBusena grafinė forma

```
relation relationBusena{
  domain{(PIM.Busena)[pavadinimas pv]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
  }
}
```

### 36. pav. Ryšio relationBusena tekstinė forma

```
mapping mappingBusena{
  domain{(PIM.Busena)[pavadinimas pv]}
  body{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
  }
}
```

### 37. pav. Ryšio realizacijos mappingBusena tekstinė forma

## 5.3.6. Elemento Agregatas transformacija

PIM metamodelio elementas Agregatas turi šiuos elementus:

- Pavadinimas
- Aprasymas
- Sujungimas
- Operacija
- Pranesimas
- Eile
- Busena

Šie elementai yra susiejami su PSM metamodelio elementais:

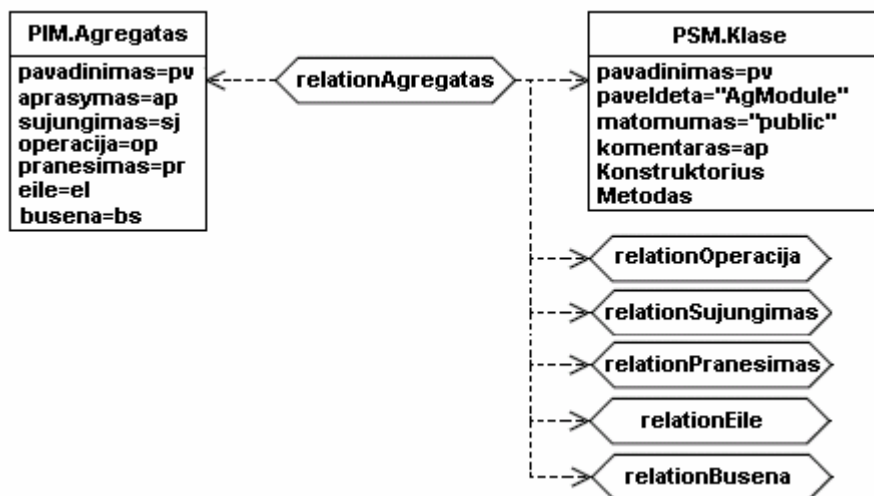
- Klase

Tam yra sudaromas ryšys relationBusena, kuris susieja šiuos metamodelių elementus. Šiame ryšyje taip pat yra panaudojami jau sudaryti atskirų elementų ryšiai:

- relationOperacija
- relationSujungimas
- relationPranesimas
- relationEile
- relationBusena



Žemiau esančiame paveiksluose pateikiama grafinė ryšio forma:



38. pav. Ryšio relationAgregatas grafinė forma

Tekstinė transformacijos forma pateikta prieduose.

#### 5.4. QVT transformacijos apibendrinimas

QVT transformacijos aprašas daug paprastesnis ir aiškesnis. Skirtingai negu XSLT, QVT transformaciją galima aiškiai suskaidyti į atskiras dalis, tuo sudarant galimybę ją pakartotinai panaudoti aprašant kitas transformacijas. QVT transformuoja pačius modelio elementus, visiškai nenaudodama XML dokumento, todėl nėra svarbu kokių formatu modeliai išsaugoti. Transformacijos aprašas nėra didelis, todėl labiau tinka didelių modelių transformacijų aprašymui negu XSLT. QVT trūkumas yra tai, kad tai dar nėra standartizuota transformavimo kalba, todėl šiuo metu dar nelabai panaudojama.

## 6. IŠVADOS

1. Iširtos modelių PIM į PSM transformacijos, nustatyta kad šiuo metu nėra standartizuotos modelių transformavimo kalbos MDA architektūroje.
2. Aliekant XSLT ir QVT transformacijų palyginimus buvo sukurta:
  - Agregatinės sistemos nuo platformos nepriklausantis PIM metamodelis.
  - Agregatinės sistemos nuo platformos nepriklausantis PIM modelis.
  - Agregatinės sistemos Java platformai skirtas PSM metamodelis.
  - EMF programavimo aplinkos priemonėmis sukurti modelių redaktoriai skirti kurti modelius ir saugoti juos XMI formatu.
  - XSLT transformacija iš PIM modelio į PSM modelį.
  - QVT transformacija iš PIM modelio į PSM modelį.
3. Aprašytos XSLT ir QVT modelių PIM į PSM transformacijos, nustatyti šie jų privalumai ir trūkumai:
  - XSLT privalumai
    - Tai standartinė XML dokumentų transformavimo kalba.
    - Yra sukurta daug priemonių, skirtų kurti XSLT transformacijas.
    - Lengvai apsirašo nedidelių modelių transformacijos.
  - QVT privalumai
    - Transformuoja ne XML dokumentą bet patį modelį.
    - Nėra priklausoma nuo modelio išsaugojimo formato.
    - Paprastesnis ir aiškesnis transformacijos aprašymas negu XSLT.
    - Dideliems modeliams transformacijų rašymas paprastesnis negu XSLT.
    - Sudaryta iš atskirų komponentų, todėl nesunku pakartotinai panaudoti kitoms transformacijos aprašyti.
  - XSLT trūkumai
    - Gali transformuoti tik XML struktūra išsaugotus modelius.
    - Sudėtingesnis transformacijos aprašymas negu QVT.
    - Sudėtinga patiems rašyti transformacijas dideliems modeliams. Tai gali būti atliekama tik naudojant pagalbines priemones.

- Sudėtingas pakartotinis transformacijos panaudojimas kitoms transformacijoms aprašyti. Sunkiai skaidosi į atskirus komponentus.
- QVT trūkumai
  - Dar nėra standartizuota.

## 7. LITERATŪRA

1. Model Driven Architecture (MDA). OMG. 2001 [žiūrēta 2005-05]. Prieiga per internetą: <http://www.omg.org/docs/ormsc/01-07-01.pdf>.
2. Kleppe A., Warmer J., Bast W. MDA Explained. The Model Driven Architecture: Practice and Promise. Boston: Addison-Wesley, 2003.
3. MDA Guide Version 1.0.1. OMG. 2003 [žiūrēta 2005-05]. Prieiga per internetą: <http://www.omg.org/docs/omg/03-06-01.pdf>.
4. Request for Proposal: MOF 2.0 Query / Views / Transformations RFP. OMG. 2002 [žiūrēta 2005-05]. Prieiga per internetą: <http://www.omg.org/docs/ad/02-04-10.pdf>.
5. XML Metadata Interchange (XMI) Specification. OMG. 2003 [žiūrēta 2005-05]. Prieiga per internetą: <http://www.omg.org/cgi-bin/apps/doc?formal/03-05-02.pdf>.
6. Jean B'ezivin, Nicolas Farcet, Jean-Marc J'ez'equel, Benoit Langlois, Damien Pollet Reflective Model Driven Engineering. [žiūrēta 2005-05]. Prieiga per internetą: <http://www.irisa.fr/triskell/publis/2003/Bezivin03.pdf>.
7. XSL Transformations (XSLT). W3C. [žiūrēta 2005-05]. Prieiga per internetą: <http://www.w3.org/TR/xslt>
8. Eclipse Modeling Framework (EMF) [žiūrēta 2005-05]. Prieiga per internetą: <http://www.eclipse.org/emf/>
9. Revised submission for MOF 2.0 Query / Views / Transformations RFP. QVT-Partners [žiūrēta 2005-05]. Prieiga per internetą: <http://qvtp.org/downloads/1.1/qvtpartners1.1.pdf>
10. QVT: The high level scope. QVT-Partners [žiūrēta 2005-05]. Prieiga per internetą: <http://qvtp.org/downloads/qvtscope.pdf>

## 8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

7 lent. Terminai

Lietuviškai	Angliškai	Paaiškinimas
Modelis	Model	Modelis yra sistemos dalies (funkcijos, struktūros ir/arba elgesio) atvaizdavimas kokioje nors sintaksiškai formalioje kalboje (pvz. UML, MOF).
Nuo platformos nepriklausomas modelis	Platform Independent Model (PIM)	PIM yra sistemos formalus modelis į kurį nėra įtrauktos realizacijos detalės
Platformos specifinis modelis	Platform Specific Model (PSM)	PSM yra sistemos formalus modelis, kuris atspindi kaip sistema yra realizuota konkrečioje platformoje.

8 lent. Santrumpos

Santrumpa	Pilnas pavadinimas	Paaiškinimas/Vertimas
MDA	Model Driven Architecture	Modeliais pagrįsta architektūra
PIM	Platform Independent Model	Nuo platformos nepriklausomas modelis.
PSM	Platform Specific Model	Platformos specifinis modelis
XMI	XML Metadata Interchange	XMI formatas
OMG	Object Management Group	OMG grupė
EMF	Eclipse Modeling Framework	Modeliavimo priemonės veikiančios Eclipse platformoje
XSLT	Extensible Stylesheet Language Transformations	XSLT transformacijų kalba
QVT	Queries/Views/Transformations	QVT transformacijų kalba

## 9. PRIEDAI

### 9.1. PIM metamodelio aprašas XMI formatu

```
<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="pim_metamodel"
  nsURI="http://pim_metamodel.ecore" nsPrefix="pim_metamodel">
  <eClassifiers xsi:type="ecore:EClass" name="Modelis">
    <eStructuralFeatures xsi:type="ecore:EReference" name="Agregatas" upperBound="-1"
      eType="##/Agregatas" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Agregatas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="aprasymas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Sujungimas" upperBound="-1"
      eType="##/Sujungimas" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Operacija" upperBound="-1"
      eType="##/Operacija" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Pranesimas" upperBound="-1"
      eType="##/Pranesimas" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Eile" eType="##/Eile"
containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Busena" eType="##/Busena"
containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Operacija">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="tipas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EIntegerObject"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="kiekis" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EIntegerObject"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Sujungimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="aprasymas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="pabaiga" lowerBound="1"
      eType="##/Agregatas"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Pranesimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Eile">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="pranesimai" eType="##/Pranesimas"
containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Busena">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="parametrai" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EIntegerObject"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
http://www.eclipse.org/emf/2002/Ecore#/EString"/>
  </eClassifiers>
</ecore:EPackage>
```

## 9.2. PIM modelio aprašas XML formatu

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE Modelis []>
<Modelis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///D:/Saulius/Mokslai/Magistratura/Magistrinis/mano/Agregatai
/PIM_Model/PIM_Model.xsd">

  <Agregatas pavadinimas="Agregatas1" aprasymas="Agregato aprasymas">
    <Sujungimas pavadinimas="Sujungimas1" pabaiga="Agregatas2" aprasymas="Sujungimo aprasymas"/>
    <Sujungimas pavadinimas="Sujungimas2" pabaiga="Agregatas3" aprasymas="Sujungimo aprasymas"/>
    <Pranesimas pavadinimas="Pranesimas1"/>
    <Pranesimas pavadinimas="Pranesimas2"/>
    <Busena pavadinimas="Busena1"/>
  </Agregatas>

  <Agregatas pavadinimas="Agregatas2" aprasymas="Agregato aprasymas">
    <Operacija pavadinimas="Operacija1" tipas="AgOpExp" kiekis="1"/>
    <Eile pavadinimas="Eile1"/>
    <Busena pavadinimas="Busena2"/>
  </Agregatas>

  <Agregatas pavadinimas="Agregatas3" aprasymas="Agregato aprasymas">
    <Sujungimas pavadinimas="Sujungimas3" pabaiga="Agregatas4" aprasymas="Sujungimo aprasymas"/>
    <Sujungimas pavadinimas="Sujungimas4" pabaiga="Agregatas1" aprasymas="Sujungimo aprasymas"/>
    <Operacija pavadinimas="Operacija2" tipas="AgOpFix" kiekis="1"/>
    <Operacija pavadinimas="Operacija3" tipas="AgOpFix" kiekis="1"/>
    <Pranesimas pavadinimas="Pranesimas3"/>
    <Pranesimas pavadinimas="Pranesimas4"/>
    <Busena pavadinimas="Busena3"/>
  </Agregatas>

  <Agregatas pavadinimas="Agregatas4" aprasymas="Agregato aprasymas">
    <Busena pavadinimas="Busena4"/>
  </Agregatas>

</Modelis>
```

### 9.3. PSM metamodelio aprašas XMI formatu

```
<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="psm_metamodel"
  nsURI="http://psm_metamodel.ecore" nsPrefix="psm_metamodel">
  <eClassifiers xsi:type="ecore:EClass" name="Sistema">
    <eStructuralFeatures xsi:type="ecore:EReference" name="Klase" upperBound="-1"
      eType="#//Klase" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Klase">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="paveldeta" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="matomumas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="komentaras" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Kintamasis" upperBound="-1"
      eType="#//Kintamasis" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Konstruktorius" upperBound="-1"
      eType="#//Konstruktorius" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Metodas" upperBound="-1"
      eType="#//Metodas" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Kintamasis">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="tipas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="matomumas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="komentaras" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Konstruktorius" eSuperTypes="#//Metodas">
    <eStructuralFeatures xsi:type="ecore:EReference" name="Metodas" upperBound="-1"
      eType="#//Metodas"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Metodas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="matomumas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="grazinamaReiksme"
      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Tekstas" upperBound="-1"
      eType="#//Tekstas" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Parametras" upperBound="-1"
      eType="#//Parametras" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Tekstas">
    <eStructuralFeatures xsi:type="ecore:EReference" name="KintamojoSukurimas" upperBound="-1"
      eType="#//KintamojoSukurimas" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="MetodoIskvietimas" upperBound="-1"
      eType="#//MetodoIskvietimas" containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Kintamasis" upperBound="-1"
      eType="#//Kintamasis" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="ObjektoSukurimas"
    eSuperTypes="#//KintamojoSukurimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="klase" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Parametras" upperBound="-1"
      eType="#//Parametras" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="Parametras">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="tipas" eType="ecore:EDatatype
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
</ecore:EPackage>
```



```

    <eStructuralFeatures xsi:type="ecore:EAttribute" name="reiksme" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="ObjektoMetodoIskvietimas"
eSuperTypes="#//MetodoIskvietimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="objektas" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="KlasesMetodoIskvietimas"
eSuperTypes="#//MetodoIskvietimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="klase" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="MetodoIskvietimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="Parametras" upperBound="-1"
eType="#//Parametras" containment="true"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="KintamojoSukurimas">
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="pavadinimas" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>
</ecore:EPackage>

```

## 9.4. Java kodo atitinkančio PSM modelio klasę Agregatas1 pavyzdys

```
/**
 * Agregatas1.java
 */

import lt.ktu.vik.jpranas.*;
import lt.ktu.vik.jpranas.jsimulation.*;

/**
 * agregato aprasymas
 */
public class Agregatas1 extends AgModule
{
    /**
     * Default constructor
     */
    public Agregatas1(AgSystem sm)
    {
        super(sm, "Agregatas1");

        Pranesimas1 = new AgMessage();
        Pranesimas2 = new AgMessage();

        Sujungimas1 = new AgItrPoint(this);
        Sujungimas2 = new AgItrPoint(this);

        Sujungimas1.connect("Agregatas2");
        Sujungimas2.connect("Agregatas3");
    };

    /**
     * @param ms
     *
     * @return
     */
    public void externalEvent(AgMessage ms)
    {
    };

    /**
     * @param ev
     *
     * @return
     */
    public void internalEvent(AgInternalEvent ev)
    {
    };

    public AgItrPoint Sujungimas1;
    public AgItrPoint Sujungimas2;
    public AgMessage Pranesimas1;
    public AgMessage Pranesimas2;
    public int Busena1;
};
```

## 9.5. XSLT transformacija

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/02/xpath-functions" exclude-result-prefixes="xs fn">
  <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/Modelis">
    <Sistema>
      <xsl:attribute
name="xsi:noNamespaceSchemaLocation">D:/Saulius/Mokslai/Magistratura/Magistrinis/mano/Agregatai/PSM_Model/PSM_Model.xsd</xsl:attribute>
      <xsl:for-each select="Agregatas">
        <Klase>
          <xsl:for-each select="@aprasymas">
            <xsl:attribute name="komentaras">
              <xsl:value-of select="."/>
            </xsl:attribute>
          </xsl:for-each>
          <xsl:for-each select="@pavadinimas">
            <xsl:attribute name="pavadinimas">
              <xsl:value-of select="."/>
            </xsl:attribute>
          </xsl:for-each>
          <xsl:attribute name="paveldeta">
            <xsl:text>AgModule</xsl:text>
          </xsl:attribute>
          <xsl:attribute name="matomumas">
            <xsl:text>public</xsl:text>
          </xsl:attribute>
          <xsl:for-each select="Sujungimas">
            <Kintamasis>
              <xsl:for-each select="@aprasymas">
                <xsl:attribute name="komentaras">
                  <xsl:value-of select="."/>
                </xsl:attribute>
              </xsl:for-each>
              <xsl:for-each select="@pavadinimas">
                <xsl:attribute name="pavadinimas">
                  <xsl:value-of select="."/>
                </xsl:attribute>
              </xsl:for-each>
              <xsl:attribute name="tipas">
                <xsl:text>AgItrPoint</xsl:text>
              </xsl:attribute>
              <xsl:attribute name="matomumas">
                <xsl:text>protected</xsl:text>
              </xsl:attribute>
            </Kintamasis>
          </xsl:for-each>
          <xsl:for-each select="Busena">
            <Kintamasis>
              <xsl:for-each select="@pavadinimas">
                <xsl:attribute name="pavadinimas">
                  <xsl:value-of select="."/>
                </xsl:attribute>
              </xsl:for-each>
              <xsl:attribute name="tipas">
                <xsl:text>Integer</xsl:text>
              </xsl:attribute>
              <xsl:attribute name="matomumas">
                <xsl:text>protected</xsl:text>
              </xsl:attribute>
            </Kintamasis>
          </xsl:for-each>
          <xsl:for-each select="Pranesimas">
            <Kintamasis>
              <xsl:for-each select="@pavadinimas">
                <xsl:attribute name="pavadinimas">
                  <xsl:value-of select="."/>
                </xsl:attribute>
              </xsl:for-each>
              <xsl:attribute name="tipas">
                <xsl:text>AgMessage</xsl:text>
              </xsl:attribute>
            </Kintamasis>
          </xsl:for-each>
        </Klase>
      </xsl:for-each>
    </Sistema>
  </xsl:template>
</xsl:stylesheet>
```

```

        </xsl:attribute>
        <xsl:attribute name="matomumas">
          <xsl:text>protected</xsl:text>
        </xsl:attribute>
      </Kintamasis>
    </xsl:for-each>
    <xsl:for-each select="Eile">
      <Kintamasis>
        <xsl:for-each select="@pavadinimas">
          <xsl:attribute name="pavadinimas">
            <xsl:value-of select="."/>
          </xsl:attribute>
        </xsl:for-each>
        <xsl:attribute name="tipas">
          <xsl:text>AgQueue</xsl:text>
        </xsl:attribute>
        <xsl:attribute name="matomumas">
          <xsl:text>protected</xsl:text>
        </xsl:attribute>
      </Kintamasis>
    </xsl:for-each>
    <xsl:for-each select="Operacija">
      <Kintamasis>
        <xsl:for-each select="@pavadinimas">
          <xsl:attribute name="pavadinimas">
            <xsl:value-of select="."/>
          </xsl:attribute>
        </xsl:for-each>
        <xsl:for-each select="@tipas">
          <xsl:attribute name="tipas">
            <xsl:value-of select="."/>
          </xsl:attribute>
        </xsl:for-each>
        <xsl:attribute name="matomumas">
          <xsl:text>protected</xsl:text>
        </xsl:attribute>
      </Kintamasis>
    </xsl:for-each>
    <Konstruktorius>
      <xsl:for-each select="@pavadinimas">
        <xsl:attribute name="pavadinimas">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <xsl:attribute name="matomumas">
        <xsl:text>public</xsl:text>
      </xsl:attribute>
      <Parametras>
        <xsl:attribute name="pavadinimas">
          <xsl:text>sm</xsl:text>
        </xsl:attribute>
        <xsl:attribute name="tipas">
          <xsl:text>AgSystem</xsl:text>
        </xsl:attribute>
      </Parametras>
      <Tekstas>
        <MetodoIskvietimas>
          <xsl:attribute name="pavadinimas">
            <xsl:text>super</xsl:text>
          </xsl:attribute>
          <Parametras>
            <xsl:attribute name="pavadinimas">
              <xsl:text>sm</xsl:text>
            </xsl:attribute>
          </Parametras>
          <Parametras>
            <xsl:for-each select="@pavadinimas">
              <xsl:attribute name="reiksme">
                <xsl:value-of select="."/>
              </xsl:attribute>
            </xsl:for-each>
          </Parametras>
        </MetodoIskvietimas>
        <xsl:for-each select="Sujungimas">
          <ObjektoSukurimas>
            <xsl:for-each select="@pavadinimas">

```

```

        <xsl:attribute name="pavadinimas">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <xsl:attribute name="klase">
        <xsl:text>AgItrPoint</xsl:text>
      </xsl:attribute>
      <Parametras>
        <xsl:attribute name="pavadinimas">
          <xsl:text>this</xsl:text>
        </xsl:attribute>
      </Parametras>
    </ObjektoSukurimas>
  </xsl:for-each>
  <xsl:for-each select="Pranesimas">
    <ObjektoSukurimas>
      <xsl:for-each select="@pavadinimas">
        <xsl:attribute name="pavadinimas">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <xsl:attribute name="klase">
        <xsl:text>AgMessage</xsl:text>
      </xsl:attribute>
    </ObjektoSukurimas>
  </xsl:for-each>
  <xsl:for-each select="Eile">
    <ObjektoSukurimas>
      <xsl:for-each select="@pavadinimas">
        <xsl:attribute name="pavadinimas">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <xsl:attribute name="klase">
        <xsl:text>AgQueue</xsl:text>
      </xsl:attribute>
    </ObjektoSukurimas>
  </xsl:for-each>
  <xsl:for-each select="Operacija">
    <ObjektoSukurimas>
      <xsl:for-each select="@tipas">
        <xsl:attribute name="klase">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <xsl:for-each select="@pavadinimas">
        <xsl:attribute name="pavadinimas">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <Parametras>
        <xsl:attribute name="reiksme">
          <xsl:text>this</xsl:text>
        </xsl:attribute>
      </Parametras>
      <Parametras>
        <xsl:attribute name="reiksme">
          <xsl:text>1</xsl:text>
        </xsl:attribute>
      </Parametras>
    </ObjektoSukurimas>
  </xsl:for-each>
  <xsl:for-each select="Sujungimas">
    <ObjektoMetodoIskvietimas>
      <xsl:attribute name="pavadinimas">
        <xsl:text>connect</xsl:text>
      </xsl:attribute>
      <xsl:for-each select="@pavadinimas">
        <xsl:attribute name="objektas">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <Parametras>
        <xsl:for-each select="@pabaiga">
          <xsl:attribute name="reiksme">
            <xsl:value-of select="."/>
          </xsl:attribute>
        </xsl:for-each>
      </Parametras>
    </ObjektoMetodoIskvietimas>
  </xsl:for-each>

```

```

        </xsl:attribute>
    </xsl:for-each>
</Parametras>
</ObjektoMetodoIskvietimas>
</xsl:for-each>
</Tekstas>
</Konstruktorius>
<Metodas>
    <xsl:attribute name="pavadinimas">
        <xsl:text>internalEvent</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="matomumas">
        <xsl:text>public</xsl:text>
    </xsl:attribute>
    <Parametras>
        <xsl:attribute name="pavadinimas">
            <xsl:text>ev</xsl:text>
        </xsl:attribute>
        <xsl:attribute name="tipas">
            <xsl:text>AgInternalEvent</xsl:text>
        </xsl:attribute>
    </Parametras>
    <Tekstas/>
</Metodas>
<Metodas>
    <xsl:attribute name="pavadinimas">
        <xsl:text>externalEvent</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="matomumas">
        <xsl:text>public</xsl:text>
    </xsl:attribute>
    <Parametras>
        <xsl:attribute name="pavadinimas">
            <xsl:text>ms</xsl:text>
        </xsl:attribute>
        <xsl:attribute name="tipas">
            <xsl:text>AgMessage</xsl:text>
        </xsl:attribute>
    </Parametras>
    <Tekstas/>
</Metodas>
</Klase>
</xsl:for-each>
</Sistema>
</xsl:template>
</xsl:stylesheet>

```

## 9.6. PSM modelio aprašas XML formatu

```
<?xml version="1.0" encoding="UTF-8"?>
<Sistema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Klase pavadinimas="Agregatas1" paveldeta="AgModule" matomumas="public" komentaras="Agregato
  aprasymas">
    <Kintamasis pavadinimas="Pranesimas1" tipas="AgMessage" matomumas="protected"/>
    <Kintamasis pavadinimas="Pranesimas2" tipas="AgMessage" matomumas="protected"/>
    <Kintamasis pavadinimas="Sujungimas1" tipas="AgItrPoint" matomumas="protected"
    komentaras="Sujungimo aprasymas"/>
    <Kintamasis pavadinimas="Sujungimas2" tipas="AgItrPoint" matomumas="protected"
    komentaras="Sujungimo aprasymas"/>
    <Kintamasis pavadinimas="Busenal" tipas="Integer" matomumas="protected"/>
    <Konstruktorius pavadinimas="Agregatas1" matomumas="public">
      <Parametras pavadinimas="sm" tipas="AgSystem"/>
      <Tekstas>
        <MetodoIskvietimas pavadinimas="super">
          <Parametras pavadinimas="sm"/>
          <Parametras reiksme="Agregatas1"/>
        </MetodoIskvietimas>
        <ObjektoSukurimas pavadinimas="Pranesimas1" klase="AgMessage"/>
        <ObjektoSukurimas pavadinimas="Pranesimas2" klase="AgMessage"/>
        <ObjektoSukurimas pavadinimas="Sujungimas1" klase="AgItrPoint">
          <Parametras pavadinimas="this"/>
        </ObjektoSukurimas>
        <ObjektoSukurimas pavadinimas="Sujungimas2" klase="AgItrPoint">
          <Parametras pavadinimas="this"/>
        </ObjektoSukurimas>
        <ObjektoMetodoIskvietimas pavadinimas="connect" objektas="Sujungimas1">
          <Parametras reiksme="Agregatas2"/>
        </ObjektoMetodoIskvietimas>
        <ObjektoMetodoIskvietimas pavadinimas="connect" objektas="Sujungimas2">
          <Parametras reiksme="Agregatas3"/>
        </ObjektoMetodoIskvietimas>
      </Tekstas>
    </Konstruktorius>
    <Metodas pavadinimas="internalEvent" matomumas="public">
      <Parametras pavadinimas="ev" tipas="AgInternalEvent"/>
      <Tekstas/>
    </Metodas>
    <Metodas pavadinimas="externalEvent" matomumas="public">
      <Parametras pavadinimas="ms" tipas="AgMessage"/>
      <Tekstas/>
    </Metodas>
  </Klase>

  <Klase pavadinimas="Agregatas2" paveldeta="AgModule" matomumas="public" komentaras="Agregato
  aprasymas">
    <Kintamasis pavadinimas="Operacijal" tipas="AgOpExp" matomumas="protected"/>
    <Kintamasis pavadinimas="Eile1" tipas="AgQueue" matomumas="protected"/>
    <Kintamasis pavadinimas="Busena2" tipas="Integer" matomumas="protected"/>
    <Konstruktorius pavadinimas="Agregatas2" matomumas="public">
      <Parametras pavadinimas="sm" tipas="AgSystem"/>
      <Tekstas>
        <MetodoIskvietimas pavadinimas="super">
          <Parametras reiksme="sm"/>
          <Parametras reiksme="Agregatas2"/>
        </MetodoIskvietimas>
        <ObjektoSukurimas pavadinimas="Operacija1" klase="AgOpExp">
          <Parametras reiksme="this"/>
          <Parametras reiksme="1"/>
        </ObjektoSukurimas>
        <ObjektoSukurimas pavadinimas="Eile1" klase="AgQueue"/>
      </Tekstas>
    </Konstruktorius>
    <Metodas pavadinimas="internalEvent" matomumas="public">
      <Parametras pavadinimas="ev" tipas="AgInternalEvent"/>
      <Tekstas/>
    </Metodas>
    <Metodas pavadinimas="externalEvent" matomumas="public">
```

```

        <Parametras pavadinimas="ms" tipas="AgMessage"/>
        <Tekstas/>
    </Metodas>
</Klase>

<Klase pavadinimas="Agregatas3" paveldeta="AgModule" komentaras="Agregato aprasymas">
    <Kintamasis pavadinimas="Pranesimas3" tipas="AgMessage" matomumas="protected"/>
    <Kintamasis pavadinimas="Pranesimas4" tipas="AgMessage" matomumas="protected"/>
    <Kintamasis pavadinimas="Operacija2" tipas="AgOpFix" matomumas="protected"/>
    <Kintamasis pavadinimas="Operacija3" tipas="AgOpFix" matomumas="protected"/>
    <Kintamasis pavadinimas="Sujungimas3" tipas="AgItrPoint" matomumas="protected"
komentaras="Sujungimo aprasymas"/>
    <Kintamasis pavadinimas="Sujungimas4" tipas="AgItrPoint" matomumas="protected"
komentaras="Sujungimo aprasymas"/>
    <Kintamasis pavadinimas="Busena3" tipas="Integer" matomumas="protected"/>
    <Konstruktorius pavadinimas="Agregatas3" matomumas="public">
        <Parametras pavadinimas="sm" tipas="AgSystem"/>
        <Tekstas>
            <MetodoIskvietimas pavadinimas="super">
                <Parametras reiksme="sm"/>
                <Parametras reiksme="Agregatas3"/>
            </MetodoIskvietimas>
            <ObjektoSukurimas pavadinimas="Pranesimas3" klase="AgMessage"/>
            <ObjektoSukurimas pavadinimas="Pranesimas4" klase="AgMessage"/>
            <ObjektoSukurimas pavadinimas="Operacija2" klase="AgOpFix">
                <Parametras reiksme="this"/>
                <Parametras reiksme="1"/>
            </ObjektoSukurimas>
            <ObjektoSukurimas pavadinimas="Operacija3" klase="AgOpFix">
                <Parametras reiksme="this"/>
                <Parametras reiksme="1"/>
            </ObjektoSukurimas>
            <ObjektoSukurimas pavadinimas="Sujungimas3" klase="AgItrPoint">
                <Parametras reiksme="this"/>
            </ObjektoSukurimas>
            <ObjektoSukurimas pavadinimas="Sujungimas4" klase="AgItrPoint">
                <Parametras reiksme="this"/>
            </ObjektoSukurimas>
            <ObjektoMetodoIskvietimas pavadinimas="connect" objektas="Sujungimas3">
                <Parametras reiksme="Agregatas4"/>
            </ObjektoMetodoIskvietimas>
            <ObjektoMetodoIskvietimas pavadinimas="connect" objektas="Sujungimas4">
                <Parametras reiksme="Agregatas1"/>
            </ObjektoMetodoIskvietimas>
        </Tekstas>
    </Konstruktorius>
    <Metodas pavadinimas="internalEvent" matomumas="public">
        <Parametras pavadinimas="ev" tipas="AgInternalEvent"/>
        <Tekstas/>
    </Metodas>
    <Metodas pavadinimas="externalEvent" matomumas="public">
        <Parametras pavadinimas="ms" tipas="AgMessage"/>
        <Tekstas/>
    </Metodas>
</Klase>

<Klase pavadinimas="Agregatas4" paveldeta="AgModule" matomumas="public" komentaras="Agregato
aprasymas">
    <Kintamasis pavadinimas="Busena4" tipas="Integer" matomumas="protected"/>
    <Konstruktorius pavadinimas="Agregatas4" matomumas="public">
        <Parametras pavadinimas="sm" tipas="AgSystem"/>
        <Tekstas>
            <MetodoIskvietimas pavadinimas="super">
                <Parametras reiksme="sm"/>
                <Parametras reiksme="Agregatas4"/>
            </MetodoIskvietimas>
        </Tekstas>
    </Konstruktorius>
    <Metodas pavadinimas="internalEvent" matomumas="public">
        <Parametras pavadinimas="ev" tipas="AgInternalEvent"/>
        <Tekstas/>
    </Metodas>
    <Metodas pavadinimas="externalEvent" matomumas="public">
        <Parametras pavadinimas="ms" tipas="AgMessage"/>
        <Tekstas/>
    </Metodas>

```



```
</Klase>  
</Sistema>
```

## 9.7. QVT transformacijos aprašas

```
relation relationAgregatas{
  domain {(PIM.Agregatas)[
    pavadinimas pv,
    aprasymas ap,
    sujungimas sj,
    operacija op,
    pranesimas pr,
    eile el,
    busena bs]}

  domain {(PSM.Klase[
    pavadinimas = pv,
    paveldeta = "AgModule",
    matomumas = "public",
    komentaras = ap,
    (PSM.Konstruktorius)[
      pavadinimas = pv,
      matomumas = "public",
      (PSM.Parametras)[
        pavadinimas = "sm"
        tipas = "AgSystem"
      ]
    ]
    (PSM.Tekstas)[
      (PSM.MetodoIskvietimas)[
        pavadinimas = "super"
        (PSM.Parametras)[
          pavadinimas = "sm"
          tipas = "AgSystem"
        ]
      ]
      (PSM.Parametras)[
        reiksme = pv
      ]
    ]
    ]
    (PSM.ObjektoSukurimas) relationOperacija(pv, sj.aprasymas, sj.pabaiga)
    (PSM.ObjektoSukurimas) relationSujungimas(pv, sj.aprasymas, sj.pabaiga)
    (PSM.ObjektoSukurimas) relationPranesimas(pv, sj.aprasymas, sj.pabaiga)
    (PSM.ObjektoSukurimas) relationEile(pv, sj.aprasymas, sj.pabaiga)
  ]
  (PSM.Kintamasis) relationOperacija(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) relationSujungimas(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) relationPranesimas(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) relationEile(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) relationBusena(pv, sj.aprasymas, sj.pabaiga)

  {PSM.Metodas}[
    pavadinimas = "externalEvent",
    (PSM.Parametras)[
      pavadinimas = "ms"
      tipas = "AgMessage"
    ]
  ]
}

  {PSM.Metodas}[
    pavadinimas = "internalEvent",
    (PSM.Parametras)[
      pavadinimas = "ev"
      tipas = "AgInternalEvent"
    ]
  ]
}
]})}
}

relation relationOperacija{
  domain{(PIM.Operacija)[pavadinimas pv, tipas t, kiekis k]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = t, matomumas = "protected"]
    (PSM.ObjektoSukurimas)[klase = t, parametras = "this", parametras = k]
  }
}
```

```

relation relationSujungimas{
  domain{(PIM.Sujungimas)[pavadinimas pv, aprasymas ap, pabaiga pb]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgItrPoint", matomumas = "protected",
komentaras = ap]
    (PSM.ObjektoSukurimas)[klase = "AgItrPoint", parametras = "this"]
  }
}

relation relationPranesimas{
  domain{(PIM.Pranesimas)[pavadinimas pv]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgMessage", matomumas = "protected"]
    (PSM.ObjektoSukurimas)[klase = "AgMessage"]
  }
}

relation relationEile{
  domain{(PIM.Eile)[pavadinimas pv]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgQueue", matomumas = "protected"]
    (PSM.ObjektoSukurimas)[klase = "AgQueue"]
  }
}

relation relationBusena{
  domain{(PIM.Busena)[pavadinimas pv]}
  domain{
    (PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
  }
}

mapping mappingAgregatas{
  domain {(PIM.Agregatas)[
  pavadinimas pv,
  aprasymas ap,
  sujungimas sj,
  operacija op,
  pranesimas pr,
  eile el,
  busena bs]}

  body {(PSM.Klase[
  pavadinimas = pv,
  paveldeta = "AgModule",
  matomumas = "public",
  komentaras = ap,
  (PSM.Konstruktorius)[
  pavadinimas = pv,
  matomumas = "public",
  (PSM.Parametras)[
  pavadinimas = "sm"
  tipas = "AgSystem"
  ]
  (PSM.Tekstas)[
  (PSM.MetodoIskvietimas) [
  pavadinimas = "super"
  (PSM.Parametras)[
  pavadinimas = "sm"
  tipas = "AgSystem"
  ]
  (PSM.Parametras)[
  reiksme = pv
  ]
  ]
  (PSM.ObjektoSukurimas) mappingOperacija(pv, sj.aprasymas, sj.pabaiga)
  (PSM.ObjektoSukurimas) mappingSujungimas(pv, sj.aprasymas, sj.pabaiga)
  (PSM.ObjektoSukurimas) mappingPranesimas(pv, sj.aprasymas, sj.pabaiga)
  (PSM.ObjektoSukurimas) mappingEile(pv, sj.aprasymas, sj.pabaiga)
  ]
  (PSM.Kintamasis) mappingOperacija(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) mappingSujungimas(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) mappingPranesimas(pv, sj.aprasymas, sj.pabaiga)
  (PSM.Kintamasis) mappingEile(pv, sj.aprasymas, sj.pabaiga)
}

```

```

        (PSM.Kintamasis) mappingBusena(pv, sj.aprasymas, sj.pabaiga)

        {PSM.Metodas}[
            pavadinimas = "externalEvent",
            (PSM.Parametras)[
                pavadinimas = "ms"
                tipas = "AgMessage"
            ]
        ]

        {PSM.Metodas}[
            pavadinimas = "internalEvent",
            (PSM.Parametras)[
                pavadinimas = "ev"
                tipas = "AgInternalEvent"
            ]
        ]
    ]})
}

mapping mappingOperacija{
    domain{(PIM.Operacija)[pavadinimas pv, tipas t, kiekis k]}
    body{
        (PSM.Kintamasis)[pavadinimas = pv, tipas = t, matomumas = "protected"]
        (PSM.ObjektoSukurimas)[klase = t, parametras = "this", parametras = k]
    }
}

mapping mappingSujungimas{
    domain{(PIM.Sujungimas)[pavadinimas pv, aprasymas ap, pabaiga pb]}
    body{
        (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgItrPoint", matomumas = "protected",
        komentaras = ap]
        (PSM.ObjektoSukurimas)[klase = "AgItrPoint", parametras = "this"]
    }
}

mapping mappingPranesimas{
    domain{(PIM.Pranesimas)[pavadinimas pv]}
    body{
        (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgMessage", matomumas = "protected"]
        (PSM.ObjektoSukurimas)[klase = "AgMessage"]
    }
}

mapping mappingEile{
    domain{(PIM.Eile)[pavadinimas pv]}
    body{
        (PSM.Kintamasis)[pavadinimas = pv, tipas = "AgQueue", matomumas = "protected"]
        (PSM.ObjektoSukurimas)[klase = "AgQueue"]
    }
}

mapping mappingBusena{
    domain{(PIM.Busena)[pavadinimas pv]}
    body{
        (PSM.Kintamasis)[pavadinimas = pv, tipas = "Integer", matomumas = "protected"]
    }
}
}

```