

VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS

INFORMATIKOS KATEDRA

ŠARŪNAS ČIRVINSKAS

MAGISTRO BAIGIAMASIS DARBAS

**VEIKLOS MODELIU GRINDŽIAMO UML SEKŲ
(SEQUENCE) MODELIO GENERAVIMAS**

Leidžiama ginti _____

Magistrantas _____
(parašas)

Darbo vadovas _____
(parašas)

(darbo vadovo mokslinis laipsnis, mokslo
pedagoginis vardas, vardas ir pavardė)

Darbo įteikimo data _____

Registracijos Nr. _____

Kaunas, 2007

TURINYS

LENTELIŲ SĄRAŠAS	4
PAVEIKSLŲ SĄRAŠAS	5
SANTRUMPŲ SĄRAŠAS	6
SANTRAUKA	7
ĮVADAS	8
1. DALYKINĖS SRITIES IŠVYSTYMO LYGIO TYRIMAS	10
1.1 Žiniomis grindžiamos IS inžinerijos raidos analizė	10
1.2 Veiklos modelio vaidmuo Žiniomis grindžiamoje IS inžinerijoje	11
1.3. Svarbiausių standartų metodų ir modeliavimo kalbų susijusių su veiklos modeliavimu analizė	13
1.3.1 GERA architektūros analizė.....	14
1.3.2 CEN ENV 40 003 CIMOSA standarto analizė	15
1.3.3 CEN ENV 12 204 standarto analizė.....	17
1.3.4 UEML modeliavimo kalbos analizė.....	20
1.3.5 Gudas, Lopata Veiklos metamodelio analizė.....	21
1.3.6 Modeliavimo kalbų, metodologijų, standartų analizės apibendrinimas	25
1.4. UML taikymas IS inžinerijoje	26
1.4.1 UML diagramų apžvalga.....	27
1.4.2 Sekų diagramos metamodelis.....	28
1.4.3 Sekų diagramos generavimo VM pagrindu galimybės	29
1.5 Projektinių modelių generavimo galimybės egzistuojančiais CASE įrankiais	30
1.5.1 Oracle developer suite 10g.....	30
1.5.2 Provision Workbench v4.0.....	30
1.5.3 System Architect v8.8.13	31
1.5.4 Magic Draw UML 7.1	31
1.5.6 CASE įrankių palyginimas.....	33
TEORINĖS DALIES IŠVADOS	35
2. VEIKLOS MODELIŲ GRINDŽIAMO UML SEKŲ MODELIO GENERAVIMO GALIMYBĖS ANALIZĖ	36
2.1 Žiniomis grindžiamos IS inžinerijos IV etapo analizė	36
2.2 Veiklos žinių kaupimo posistemio analizė	37
2.3 Veiklos metamodelio analizė	38
2.4 Analizuojami UML modeliai	40
2.5 UML sąveikos modelių analizė	41

2.6. Siūlomas UML Sekų metamodelis	44
2.7 Veiklos metamodelio atvaizdavimas į sekų metamodelį	45
2.8 Veiklos metamodelio papildymas	47
3. VEIKLOS MODELIU GRINDŽIAMO UML SEKŲ MODELIO GENERAVIMAS	48
3.1 Siūloma sekų diagramos loginė duomenų bazės schema	48
3.2 Veiklos modelio loginės duomenų bazės schemos papildymas.....	49
3.3 Siūlomas UML Sekų diagramos generavimo iš Veiklos modelio algoritmas	55
3.4 Siūlomas UML Sekų modelio generavimo iš veiklos modelio programos prototipas	56
IŠVADOS	66
LITERATŪROS SĄRAŠAS	68
PRIEDAI	70
1. Priedas „Informacinės Technologijos 2006“ Konferencijos pranešimas	70

LENTELIŲ SĄRAŠAS

1lentelė Modeliavimo standartų metodologijų ir kalbų palyginimas.....	25
2lentelė Populiariausių CASE įrankių palyginimas	33
3lentelė Veiklos metamodelio sudarymo notacija	39
4lentelė UML sekų diagramos notacija.....	42
4lentelė tęsinys	43
5lentelė Veiklos metamodelio klasių atvaizdžiai į sekų metamodelį.....	45
6lentelė Veiklos modelio db lentelės Procesas detalus aprašymas	49
7lentelė Veiklos modelio db lentelės ProcAtr detalus aprašymas.....	50
8lentelė Veiklos modelio db lentelės Funkcija detalus aprašymas	50
9lentelė Veiklos modelio db lentelės FunkcijosAtrb detalus aprašymas	50
10lentelė Veiklos modelio db lentelės Vykdytojas detalus aprašymas.....	51
11lentelė Veiklos modelio db lentelės Vykduom detalus aprašymas	51
12lentelė Veiklos modelio db lentelės MatSrautas detalus aprašymas	51
13lentelė Veiklos modelio db lentelės MatsrAtrb detalus aprašymas.....	51
14lentelė Veiklos modelio db lentelės InfSrautas detalus aprašymas.....	52
15lentelė Veiklos modelio db lentelės ProcMSr detalus aprašymas.....	52
16lentelė Veiklos modelio db lentelės FunkcIsr detalus aprašymas	52
17lentelė Veiklos modelio db lentelės Ivykis detalus aprašymas	53
18lentelė Veiklos modelio db lentelės Ivyk_Proc detalus aprašymas.....	53
19lentelė Veiklos modelio db lentelės ProcFunkc detalus aprašymas.....	53
20lentelė Veiklos modelio db lentelės ProcFunkcAtributai detalus aprašymas	53
21lentelė Veiklos modelio db lentelės Infveikla detalus aprašymas.....	54
22lentelė Veiklos modelio db lentelės Aktyvacija detalus aprašymas.....	54
23lentelė Veiklos modelio db lentelės Informacinio srauto atributai detalus aprašymas	54

PAVEIKSLŲ SĄRAŠAS

1 pav. Pagrindiniai IS inžinerijos raidos etapai.....	10
2 Pav. Veiklos modelio vieta tradiciniame IS kūrimo gyvavimo cikle.....	11
3 Pav. UML diagramų generavimas naudojant Veiklos žinių posistemį.....	12
Šaltinis: Michael Petit, 2002, Report on the State of the Art in Enterprise Modelling.....	13
4 Pav. standartai reglamentuojantys veiklos modeliavimą.....	13
Šaltinis: Michael Petit, 2002, Report on the State of the Art in Enterprise Modelling.....	15
5 Pav. GERA standarto principinė modeliavimo schema.....	15
6 Pav. CIMOSA standarto principinė modeliavimo schema.....	15
7 Pav. CIMOSA standarto pagrindiniai modeliavimo konstruktai.....	16
8 Pav. CEN ENV 40 003 santykis su kitomis modeliavimo kalbomis ir standartais.....	17
9 Pav. CEN ENV 12 204 standarto principinė modeliavimo schema.....	19
10 Pav. UEML metodologijos principinė modeliavimo schema.....	21
11 Pav. CASE įrankio struktūra su žinių baze.....	23
12 Pav. Žinių bazės klasių diagrama.....	24
13 Pav. UML diagramos.....	27
Šaltinis: sukurta autoriaus.....	29
14 pav. Sekų modelio generavimo iš veiklos modelio principinė schema.....	29
15 pav. Ketvirto IS inžinerijos vystymosi etapo principinė schema.....	36
16 pav. veiklos modelio vieta žiniomis grindžiamame CASE įrankyje.....	37
17 pav. Veiklos metamodelio klasių modelis.....	38
18 pav. Analizuojamos UML diagramos.....	40
19 pav. Bendradarbiavimo diagramos pavyzdys.....	41
20 pav. Sekos diagramos pavyzdys.....	42
21 pav. Siūlomas UML sekų metamodelis.....	44
22 pav. Veiklos metamodelio atvaizdavimas į UML sąveikų metamodelį.....	46
23 pav. Papildytas veiklos metamodelis.....	47
24 Pav. Siūloma Sekų diagramos loginė duomenų bazės schema.....	48
25 pav. Veiklos modelio loginės duomenų bazės schemos papildymas.....	49
26 pav. Siūlomas UML Sekų diagramos generavimo iš Veiklos modelio algoritmas.....	55

SANTRUMPŲ SĄRAŠAS

IS- informacinė sistema

UML-unifikuota modeliavimo kalba

ISO-tarptautinė standartų organizacija

CEN-Europos standartizacijos komitetas

CASE- kompiuteriu paremtos sistemos inžinerija

DB-duomenų bazė

OMG-objektinio valdymo grupė

OO-objektiškai orientuotas

UEML-universali veiklos modeliavimo kalba

GERA –apibendrinta veiklos modeliavimo architektūra

GERAM- apibendrinta veiklos modeliavimo architektūra ir metodologija

CIMOSA- atvirų sistemų architektūra kompiuteriu paremtai gamybai

VM- veiklos modelis

VMM- veiklos metamodelis

SANTRAUKA

Šarūnas Čirvinskas. (2007) *UML sequence model generation based on enterprise model*. MBA graduation paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of informatics 75p.

Summary

Main objective: Create a method and realization algorithms for UML sequence model based on enterprise model generation.

Main tasks:

- To analyze the work domain: knowledge based IS engineering
- To implement enterprise metamodel's content analysis through the prism of UML sequence metamodel
- To create the way (algorithms) to generate UML sequence diagram.
- Suggest a software prototype for created method.
- To test the software prototype using real world data.

Research domain: design stage of knowledge based IS engineering.

Research methodology

During the research main methods used were analysis and synthesis comparing, analysis of science literature and Internet sources, model creation, analysis and decomposition.

Results:

The analysis of work domain was implemented. Enterprise model structure was analyzed through its metamodel. Enterprise metamodels structure was compared to the sequence metamodel and it was found that the enterprise metamodel is not sufficient to create a sequence diagram. Therefore the enterprise metamodel was complemented with additional concepts. The sequence diagram generation algorithm was created and tested with a software prototype.

Consequences:

After enterprise metamodel and sequence metamodel decomposition and analysis there was designated that in domain knowledge repository –enterprise model stored knowledge is insufficient to generate UML sequence diagram. Therefore the knowledge repository was complemented with additional concepts.

After research there was designated that sequence model generation using the created algorithm and software prototype intellectualizes IS engineering design stage because of the reduced error possibility and time consumption.

Paper consists of 60 pages, 26 graphs and 23 tables.

IVADAS

Tema: Veiklos modeliu grindžiamo UML sekų (angl. sequence) modelio generavimas.

Darbo problema: šiuo metu IS inžinerijoje projektiniai modeliai yra kuriami empiriškai, formaliai jų nepatikrinus pagal realią dalykinę sritį. Šiuo darbu siekiama intelektualizuoti IS inžinerijos projektavimo etapą generuojant UML sekų modelį iš dalykinės srities žinių saugyklos vadinamos veiklos modeliu.

Šio **darbo paskirtis** – sugeneruoti veiklos modeliu grindžiamą UML sekų modelį, atliekant veiklos modelio analizę. Nustatyti veiklos modelio elementus atitinkančius UML sekų modelio elementus metamodelio lygyje. Šiuolaikiniame dinamiškame pasaulyje efektyvios vidinės ir išorinės komunikacijos užtikrinimas įmonėje yra viena pagrindinių įmonės veiklos efektyvumo didinimo sąlygų. Efektyvios organizacijos komunikacinės sistemos kūrime svarbų vaidmenį vaidina įmonės veiklos modeliavimas, kurio pagrindu ir yra atvaizduojami atskirų įmonės padalinių, darbuotojų tarpusavio sąveika (angl. interaction), parodoma detali komunikacinio proceso eiga. Šiame darbe bus atliekamas UML (angl. unified modeling language) sekų modelio generavimas. Sugeneruoto sekų modelio pagalba bus intelektualizuojamas projektavimo etapas padedantis identifikuoti problemas kylančias IS kūrimo metu bei ir optimizuoti patį kūrimo procesą. Reikalinga akcentuoti tai, kad UML sekos modelio generavimas bus vykdomas, remiantis veiklos modeliu – realių įmonėje vykstančių procesų saugykla.

Darbo objektas: žiniomis grindžiamos IS inžinerijos projektavimo etapas.

Darbo tikslas: patobulinti IS inžinerijos projektavimo etapą, taikant UML sekų modelio generavimo iš veiklos modelio algoritmus.

Darbo uždaviniai:

1. Išanalizuoti dalykinę sritį –žiniomis grindžiamą IS inžinerijos projektavimo etapą.
2. Atlikti veiklos metamodelio sudėties analizę UML sekų metamodelio aspektu, siekiant nustatyti trūkstamus veiklos metamodelio komponentus.
3. Sukurti UML sekų modelio generavimo būdą (algoritmus)
4. Pasiūlyti sukurto metodo programinės įrangos prototipą
5. Patikrinti programinės įrangos prototipą taikant realius duomenis.

Darbo struktūra: Darbas susideda iš trijų pagrindinių dalių. Pirmojoje dalyje yra apibrėžiama dalykinė sritis –žiniomis grindžiama informacinių sistemų inžinerija. Analizuojami pagrindiniai metodai, modeliavimo kalbos, standartai reglamentuojantys veiklos modeliavimą. Taip pat lyginami populiariausi šiuo metu CASE įrankiai pagal projektinių modelių, kodo, dokumentacijos generavimo galimybes. Antrojoje darbo dalyje yra analizuojamas žiniomis

grindžiamos inžinerijos etapas jo struktūriniai elementai. Pateikiamas sukurtas sekų metamodelis, aprašomas jo atvaizdavimas į veiklos metamodelį bei jo papildymas. Trečioje darbo dalyje pateikiama sukurta sekų diagramos duomenų bazės loginė schema. Atliekamas veiklos modelio duomenų bazės loginės schemos pildymas naujais elementais. Pateikiamas ir realiais duomenimis patikrinamas sekų diagramos generavimo algoritmas bei programinės įrangos realizuojančios jį prototipas.

Darbe naudoti literatūros šaltiniai: Rengiant darbą buvo panaudotas 22 literatūros šaltiniai tiek Lietuvos, tiek užsienio autorių. Analizuota autorių S. Gudo, A. Lopatos, T. Skersio moksliniai darbai žiniomis grindžiamos IS inžinerijos ir veiklos modeliavimo klausimais. Taip pat naudotasi užsienio autorių internete publikuojamais moksliniais darbais, tyrimų rezultatais. Išanalizuotos keturi šiuo metu populiariausi CASE įrankių paketai remiantis jų gamintojų interneto portaluose publikuojama dokumentacija bei demonstracinėmis versijomis.

Tyrimo metodai:

1. Mokslinės literatūros ir Internetinių šaltinių analizės ir apibendrinimo metodai.
2. Palyginimo ir sintezės metodai.
3. Modelių struktūrinės analizės metodai.
4. Sekų metamodelio kūrimo metodas
5. Prototipavimo metodas.

Teorinė darbo reikšmė: Sekų modelio generavimas Intelektualizuoja IS inžinerijos projektavimo etapą, tuo kad automatizuoja sekų modelio kūrimą iš dalykinės srities žinių saugyklos –veiklos modelio.

Praktinė darbo reikšmė: Sukurtas UML sekų modelio generavimo algoritmas bei pasiūlytas programinės įrangos prototipas leidžia generuoti sekų modelį naudojant dalykinės srities žinių saugyklą- veiklos modelį.

Darbo apimtis: magistro baigiamąjį darbą sudaro septyni skyriai kurie savo ruožtu skirstomi į poskyrius. Darbe yra 75 puslapis, 26 paveikslai, 23 lentelės ir 1 priedas

Rezultatų aprobavimas:

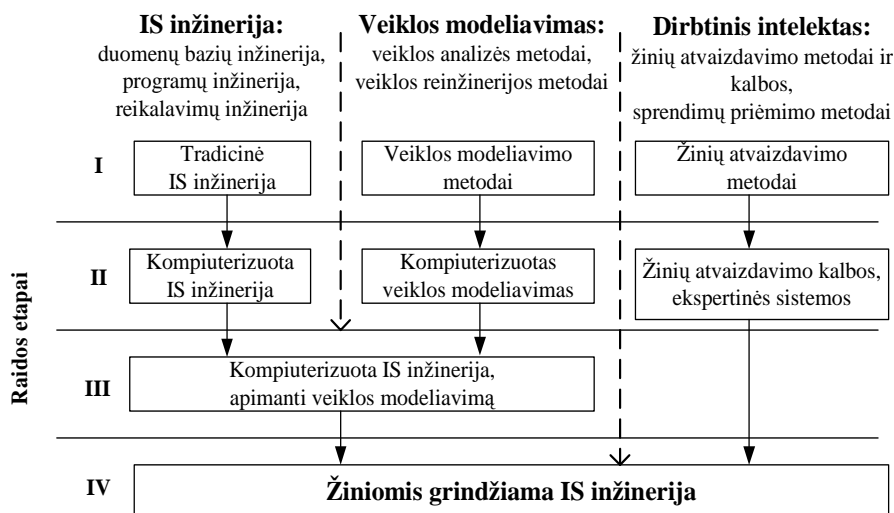
1. Parašytas ir apgintas straipsnis 11-ojoje tarpuniversitetinėje doktorantų ir magistrantų konferencijoje „Informacinės technologijos 2006“. Š. Čirvinkas, A. Lopata „Veiklos modeliu grindžiamas UML sekų modelio generavimas“.

1. DALYKINĖS SRITIES IŠVYSTYMO LYGIO TYRIMAS

Sekančiame skyriuje yra apibrėžiamas darbo objektas, analizuojami su metodai, standartai, modeliavimo kalbos turinčios įtaką darbo objektui. Tiriama populiariausi programinės įrangos paketai siekiant nustatyti esamą išvystymo lygį dalykinės srities ribose.

1.1 Žiniomis grindžiamos IS inžinerijos raidos analizė

Pastaraisiais metais kompiuterizuota IS inžinerija (CASE) yra pasiekusi naują kokybinį vystymosi lygį – ji tampa žiniomis grindžiama IS inžinerija. Reikia pastebėti, kad jau du dešimtmečiai (Gudas S., Lopata A. 2005, p. 1) informacinės sistemos yra kuriamos CASE sistemų (angl. *Computer Aided Systems Engineering tools*) aplinkose. Iš viso šiuo metu priskaičiuojama apie šimtą skirtingų CASE sistemų. Iš pradžių (žr. 1 pav.) funkcionavo tik tradicinė IS inžinerija, kurią vėliau keitė kompiuterizuota IS inžinerija. Dar vėliau į kompiuterizuotą IS inžineriją buvo integruotas veiklos modeliavimas. Galiausiai informacinių technologijų vystymasis, informacijos svarba šiuolaikiniame versle, naujų verslo modeliavimo priemonių kūrimas paskatino naują žiniomis paremtą CASE sistemą. Ketvirtajame IS inžinerijos raidos etape CASE sistemos dalimi tapo žinių posistemis, kuriame yra kaupiamos būtinos žinios apie veiklos sritį (organizacijos procesus ir funkcijas). Žinių posistemio struktūrą apibrėžia formalizuota žinių apie veiklos valdymą struktūra - veiklos metamodelis (Gudas S., Lopata A. 2005, p. 4). Dabartinė IS inžinerija integruoja kompiuterizuotą IS inžineriją, veiklos modeliavimą ir dirbtinį intelektą (žr. 1 pav.). Žiniomis pagrįsta IS inžinerija susistemino ir apibendrino anksčiau naudotus IS inžinerijos vystymo etapus ir leido informacinių sistemų projektuotojui savo veikloje papildomai naudotis veiklos žinių saugyklose sukauptą informacija. Šiame darbe bus analizuojamas vienas iš ketvirtojo IS inžinerijos etapo aspektų: sekų metamodelio generavimas remiantis turimu veiklos metamodeliu.

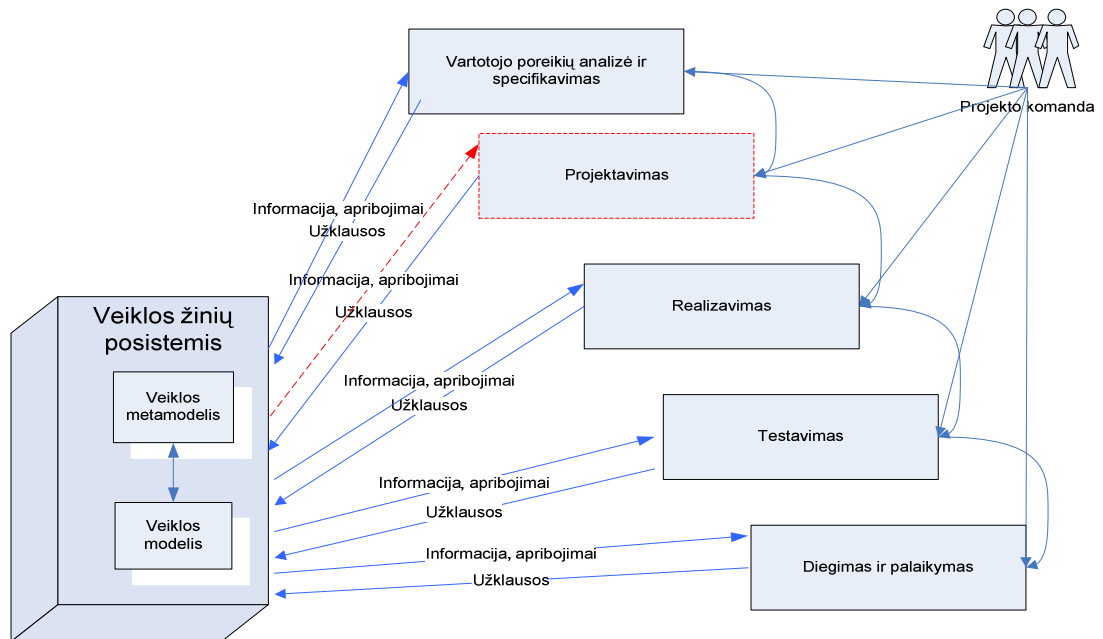


Šaltinis: Gudas S., Lopata A. (2005) Žiniomis grindžiamos informacijos sistemų inžinerijos bruožai

1pav. Pagrindiniai IS inžinerijos raidos etapai

1.2 Veiklos modelio vaidmuo Žiniomis grindžiamoje IS inžinerijoje.

Veiklos žinių kaupimo posistemį sudaro veiklos modelis ir jo struktūrą formalizuojantis veiklos metamodelis. Išsaugotos dalykinės srities žinios yra naudojamos visuose IS kūrimo gyvavimo ciklo etapuose. Žinių posistemis atlieka kontrolės bei informacinę funkcijas. Principinė veiklos modelio vietos IS kūrimo procese schema yra pateikta žemiau esančiame paveiksle

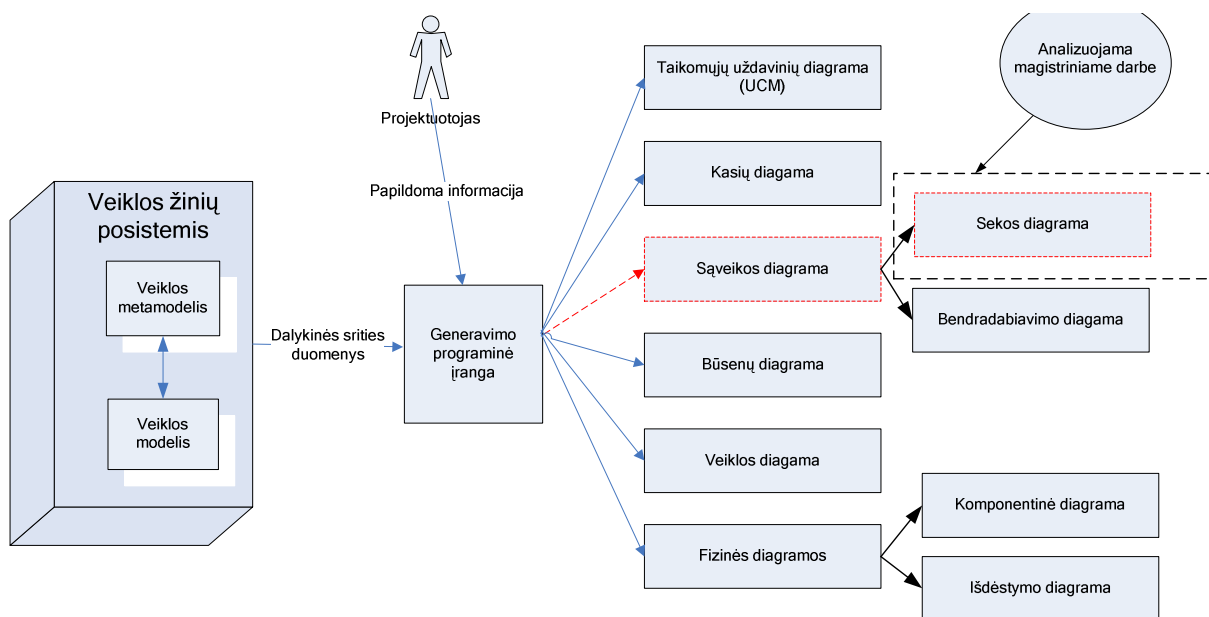


Šaltinis: sukurta autoriaus

2 Pav. Veiklos modelio vieta tradiciniame IS kūrimo gyvavimo cikle

Projektuojant IS komunikavimas su veiklos žinių posistemiū vyksta užklausių apribojimų pagalba

Veiklos modelyje saugoma dalykinės srities informacija bei metamodelyje saugomas formalus struktūros aprašymas leidžia automatizuoti UML diagramų kūrimą IS projektavimo etape. Kaip matome 3paveiksle Naudojant veiklos modelyje saugomas žinias interaktyviai dalyvaujant sistemos projektuotojui bei naudojant tinkamą generavimo programinę įrangą galima sugeneruoti, bet kurią UML diagramą.



Šaltinis: sukurta autoriaus

3 Pav. UML diagramų generavimas naudojant Veiklos žinių posistemį

Toliau darbe yra pateikiama veiklos modeliavimo standartų analizė

1.3. Svarbiausių standartų metodų ir modeliavimo kalbų susijusių su veiklos modeliavimu analizė

Šiuo metu yra keletas skirtingų organizacijos modeliavimo metodų ir kalbų IDEF, OMT, UML, CIMOSA, ARIS ir kt. Tai sąlygojo tarpusavyje nesuderinamų programinių paketų, skirtų organizacijos veiklos modeliavimui atsiradimą (ARIS ToolSet, System Architect, FirstSTEP, CimTool ir kt.). Pagrindinės organizacijos modeliavimo mokslinės grupės, tokios kaip ODP, OMG, PSL/NIST, didžiausi programinės įrangos gamintojai (Oracle, Microsoft) bei pagrindinės standartizacijos organizacijos (ISO, CEN) stengiasi sukurti bendrą standartą, kuriuo remiantis būtų kuriami nauji organizacijos veiklos modeliavimo metodai, kalbos bei su jais suderinta programinė įranga. (Lopata A. 2002) Sekančioje lentelėje pateikiami ISO CEN bei kitų organizacijų išleisti standartai reglamentuojantys veiklos modeliavimą

		A r c h i t k e t ū r a	M e t o d o l o g i j a	K a l b a	N u o r o d o s	I T I n f r a s
ISO	ISO TR 10314, Nuorodų modelis gamybinės produkcijos standartams				X	
	ISO/DIS 13281.2 -Gamybos automatizacijos programavimo aplinka					X
	ISO 14258, Industrinės automatizacijos sistemos -veiklos modelių konceptai ir taisyklės	X				
	ISO/IEC CD15288, Gyvavimo ciklo valdymo sistema, Gyvavimo ciklo procesai	X				
	ISO 15704, Reikalavimai veiklos modeliavimo nuorodoms, architektūrai ir metodologijoms	X				
	ISO FDIS 15531-1 - Gamybos valdymo duomenų apykaita				X	
	ISO 16668, Bazinis semantintis registras (BSR) -Taisyklės, nurodymai ir metodologija			X		
	ISO FCD 15414 (2000) - Atviras paskirstytas apdorojimas, nuorodų modelis			X		X
CEN	ENV 40003, CIM sistemų architektūra modeliavimo pagrindai	X				
	ENV 12204, Veiklos modeliavimo konstruktai			X		
	ENV 13550 -Veiklos modeliavimo vykdymo ir integravimo paslaugos (EMEIS)					X
Kiti	Open Group (2000), TOGAF –Atviros grupės architektūros struktūra					X
	OMG (1992), Objektų valdymo architektūra (COBRA)					X
	ISA-dS95, Amerikos instrumentų visuomenė, Veiklos kontrolės sistemos integracija				X	

Šaltinis: Michael Petit, 2002, Report on the State of the Art in Enterprise Modelling

4 Pav. standartai reglamentuojantys veiklos modeliavimą

Sekančiuose skyriuose apžvelgiami pagrindiniai veiklos modeliavimo standartai bei metodologijos.

1.3.1 GERA architektūros analizė

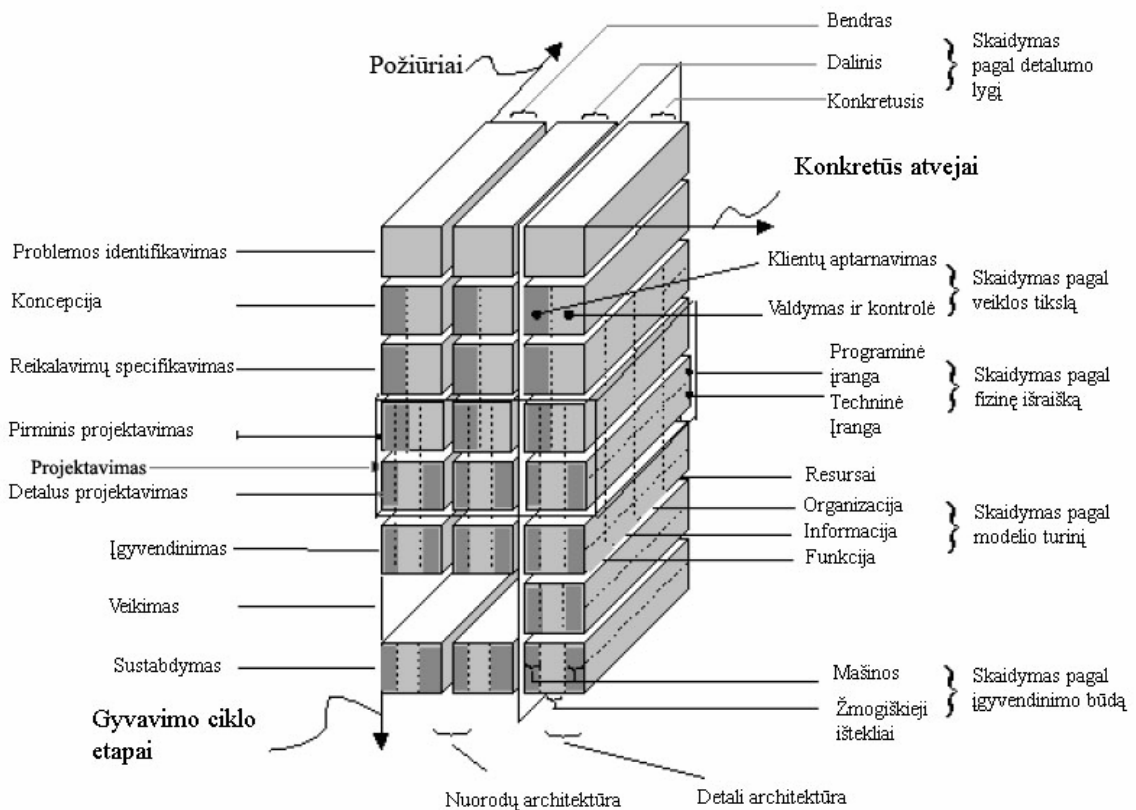
GERA (Generalised Enterprise Reference Architecture) apima visas žinias reikalingas veiklos modeliavimui bei integravimui. GERAM yra apibūdinama pragmatiniu požiūriu per apibendrintą komponentų apibrėžimo struktūrą reikalingą visuose veiklos modeliavimo etapuose.

GERA pateikia analizės ir modeliavimo struktūrą kuri yra paremta gyvavimo ciklo konceptu , ši architektūra turi tris veiklos modeliavimo apibrėžimo dimensijas.

Gyvavimo ciklo dimensiją skirtą kontroliuojamų veiklos esybių procesų pagal gyvavimo ciklo etapus modeliavimui.

Bendrumo dimensiją skirtą dekomponavimo procesui nuo bendro iki dalinio ir iki konkrečiau.

Apžvalgos dimensiją skirtą specifinių veiklos esybių sudėtinių dalių atvaizdavimui. 5 Pav. pavaizduota trijų dimensijų GERA struktūra aprašyta aukščiau. kuri atspindi šią veiklos modeliavimo architektūrą. Aprašomoji modeliavimo architektūros dalis susideda tik iš bendro ir dalinio lygio. Šie lygiai aprašo veiklos modeliavimo koncepcijas bei bendrus ir makro lygio konstruktus (modeliavimo kalbas), naudojamus veiklos aplinkai aprašyti. Konkretusis lygis atvaizduoja modeliavimo proceso rezultatus, modelį ar veiklos esybės apibūdinimą priklausantį tam tikram gyvavimo ciklo etapui. Tačiau yra siekiama kad modeliavimo kalbos palaikytų hierarchinį santykį su modeliais gretimose gyvavimo ciklo fazėse t.y. modelių išvedimą iš aukštesnės klasės į žemesnę ir atvirkščiai iš žemesnės į abstraktesnę, kad nereikėtų kurti atskirų modeli skirtingoms gyvavimo ciklo stadijoms. (Michael Petit 2002)

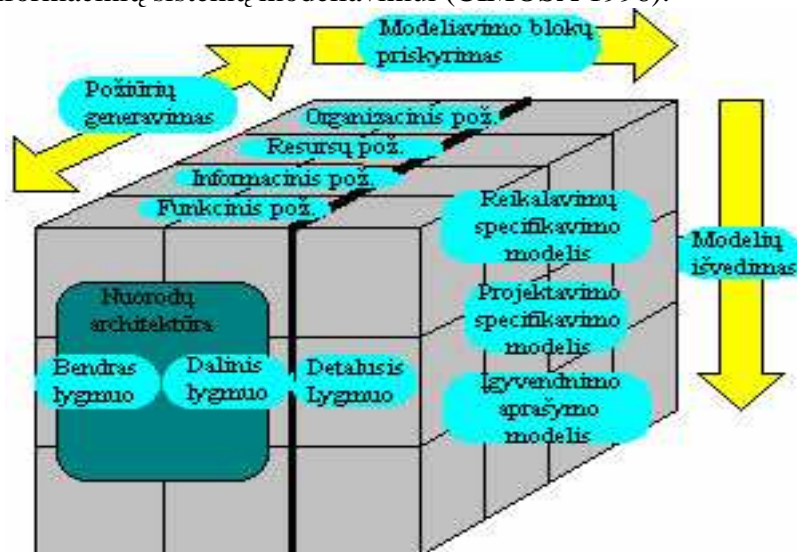


Šaltinis: Michael Petit, 2002, Report on the State of the Art in Enterprise Modelling

5 Pav. GERA standarto principinė modeliavimo schema

1.3.2 CEN ENV 40 003 CIMOSA standarto analizė

Šekančiame skyriuje aprašomas veiklos modeliavimo standartas CEN ENV 40 003 dar žinomas kaip CIMOSA (Open Systems Architecture for Computer Integrated Manufacturing), naudojamas veiklos informacinių sistemų modeliavimui (CIMOSA 1996).



Šaltinis: CIMOSA, 1996, A Primer on key concepts, purpose and business value

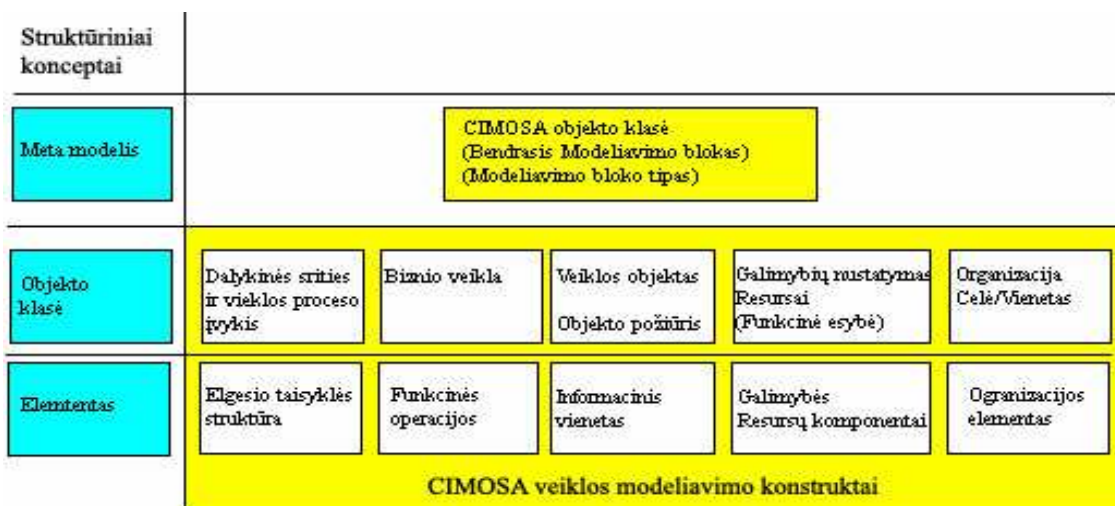
6 Pav. CIMOSA standarto principinė modeliavimo schema

Modeliavimo architektūra pavaizduota aukščiau esančiame paveiksle išskaido CIMOSA specifikavimo architektūrą į bendrą ir dalinį lygius kiekvienas lygis atspindi skirtingus tam tikro

veiklos modeliavimo požūrius. Požūrių koncepcija leidžia vartotojui dirbti su tam tikra modelio dalimi tuo sumažinant modeliavimo sudėtingumą ir susikoncentruojant į tam tikrą sritį taip geriausiai atspindint vartotojo poreikius.

CIMOSA modeliavimo architektūra palaiko trijų lygių IS inžinerijos gyvavimo ciklą (Reikalavimų specifikavimas, Modeliavimo aprašymas ir įgyvendinimo specifikavimo modeliai). IS kūrimo seka gali kisti t.y. galima pradėti kūrimą nuo bet kurio gyvavimo ciklo etapo etapai taip pat gali būti iteratyvus. Taip pat priklausomai nuo inžinerinės paskirties gali likti neįgyvendintų gyvavimo ciklo etapų.

Informacinės sistemos neturėtų būti kuriamos kaip dideli monolitiniai modeliai, efektyviau yra kurti kooperuojančių modelių rinkinį. Tokią koncepciją įgyvendina CIMOSA modeliavimo architektūra įgalindama integruotą veiklos modeliavimą, leidžiantį skirtingiems specialistams modeliuoti atskiras veiklos sritis tačiau galutinis produktas gaunamas integruotas organizacijos veiklos modelis. (CIMOSA 1996)

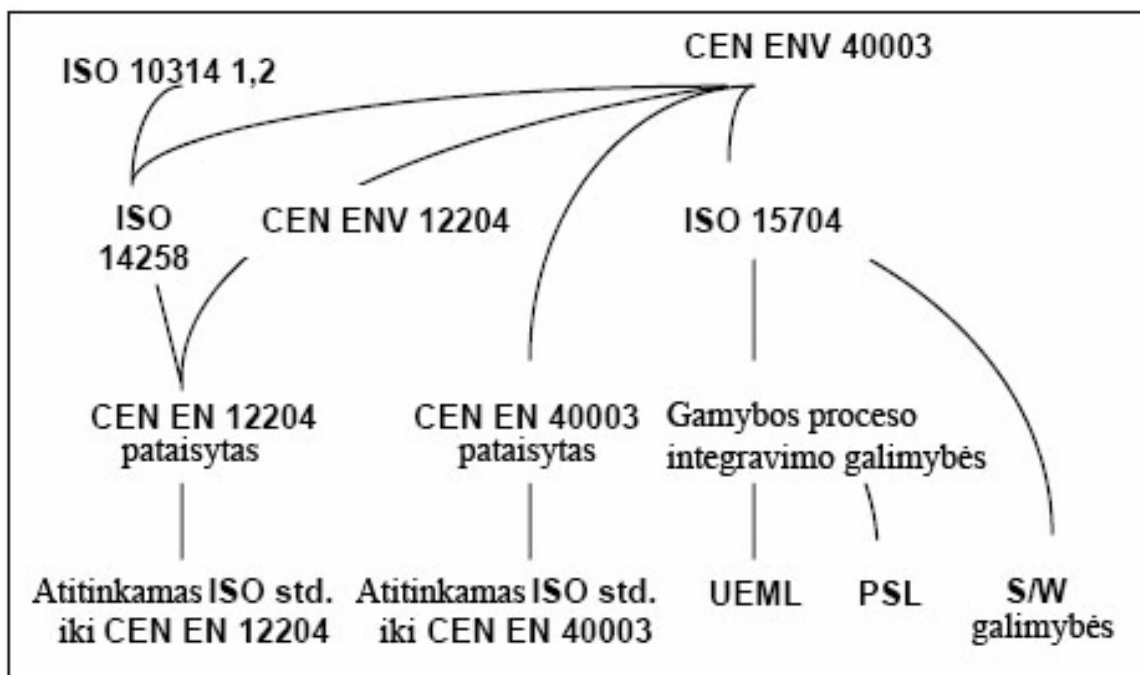


Šaltinis: CIMOSA, 1996, A Primer on key concepts, purpose and business value

7 Pav. CIMOSA standarto pagrindiniai modeliavimo konstruktai

Aukščiau esančiame paveiksle pavaizduota principinė CIMOSA konstruktų schema. Procesai įvykiai ir veiklos yra objektinės klasės, kurios aprašo organizacijos veiklos funkcijas ir elgesį (dinamiką). Veiklos procesų įeiga ir išeiga aprašo informacija ir resursus reikalingus IS inžinerijoje. Organizaciniai aspektai yra aprašomi atsakomybių ir teisių pavidalu funkcijų resursų ir organizacijos konstruktams. Jie struktūrizuojami kaip organizaciniai vienetai ar celės į objektų klasių hierarchiją.

CEN ENV 40 003 santykis su kitais standartais bei architektūromis pateiktas sekančiame paveiksle



Šaltinis: David Shorter, 2004 Modeling and Architecture Work program and key resources

8 Pav. CEN ENV 40 003 santykis su kitomis modeliavimo kalbomis ir standartais

1.3.3 CEN ENV 12 204 standarto analizė

CEN ENV 12 204 standarte yra pagrindinių konstrukčių reikalingų kompiuteriu paremtos veiklos modeliavimui apibrėžimai ir aprašai. Standarto turinys yra sutelktas į integruotas kompiuterines gamybos sistemas įskaitant valdymo, gamybos bei žmogiškųjų išteklių kontrolės užduotis. Modeliai sukurti naudojant konstruktus suderintus su šiuo standartu turi būti kompiuteriu apdorojami ir turėtų galiausiai įgalinti veiklos operacijų stebėjimą ir kontrolę. Modeliavimo kalbos kontraktai vaizduojami kaip bendrosios kalbos elementai yra skirti pagerinti vartotojų supratimą apie veiklos modeliavimą ir vystyti bendrą aprašymo kultūrą. Veiklos modeliavimo vystymosi procesas yra skirtingų CIM elementų ir aspektų transformacija į integruotus kompiuteriu apdorojamus modelius kurių kūrimą turėtų įgalinti bendri standartizuoti lengvai panaudojami modeliavimo konstruktai.

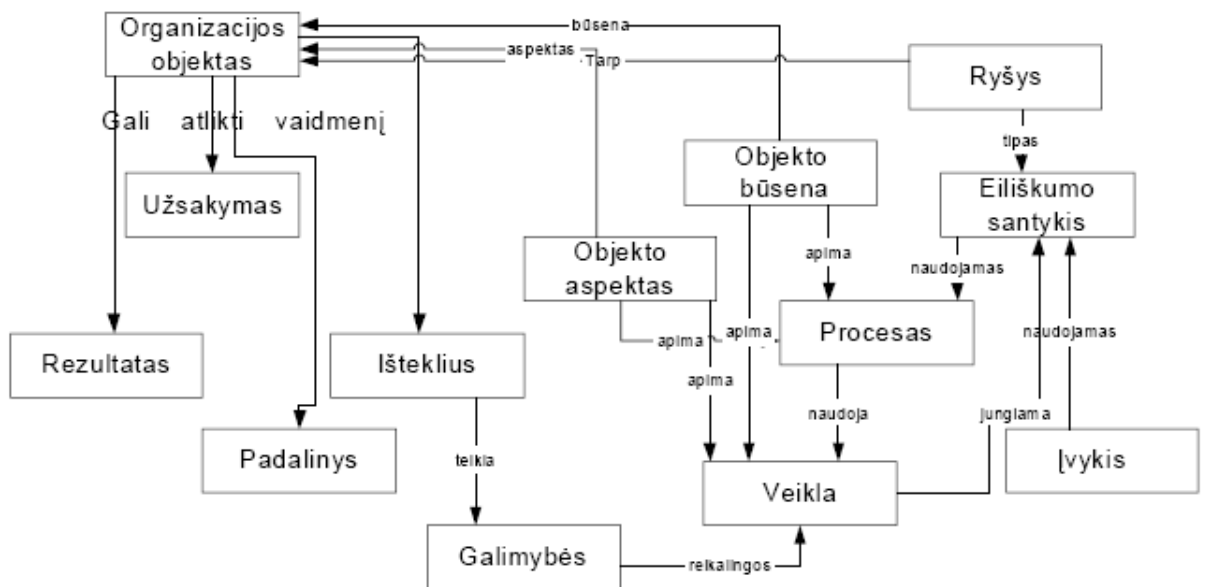
Bendri konstruktai gali būti specializuojami ir perorientuojami į struktūras specifinėms užduotims atlikti pvz.: tam tikram gamybos sektoriui ar tam tikrai veiklos problemai modeliuoti. Paeiliui tokios struktūros kartu su bendrais konstruktais turėtų būti naudojamos konkrečios organizacijos konkretiems modeliams kurti. Modeliavimo procesas yra hierarchinis bei susidedantis iš kelių dimensijų skirtingus aspektus atspindi Požiūrio dimensija, detalumo lygis atspindi Modelio dimensijoje.

Konstruktais paremtos modeliavimo kalbos išsiskiria savo privalumais tiek verslo tiek IT vartotojams:

- Padėdamos kurti modeliavimo metodu ir įrankius kurie galėtų būti naudojami daugiau gamybos specialistų nei IT personalo.
- Pagerindamos sistemos suprantamumą (reikalavimų specifikavimo, modeliavimo, ir įgyvendinimo etapuose) sujungiant sistemos elementus ir šnekamąją kalbą.
- Remdamos pagrindines keturias sistemos geros kokybės savybes (efektyvumą, suprantamumą, patikimumą, modifikuojamumą).

Taip pat galima paminėti keletą specifinių privalumų tokių kaip: greitesnį ir geresnį sprendimų priėmimo procesą, modelio komponentų kūrimą iš keleto šaltinių, greitesnį modelio kūrimą bei įgyvendinimą.

Konstruktai. Modeliavimo kalbos konstruktai susideda iš bendros architektūros iš kurios gali būti išvestas platus spektras modelių su didelėmis pritaikymo galimybėmis CIM ir veiklos modeliavime. Organizacijos priklausančios tam pačiam gamybos sektoriui dažniausia domisi tam tikrais modeliavimo konceptų (specializacijų) rinkiniais aprašomais kaip „building block“ šiame standarte. Paeiliui specifinės organizacijos dar labiau sukonkretina minėtus „building block“ ir pritaiko juos savo operacijoms. Tarkim aviacijoje turbinos mentė tipas 1 gali būti sukonkretintas lėktuvo variklio tipas. Šie sukonkretinti ir vartotojo aprašyti „building block“ sudaro architektūrą tinkamą tam tikram gamybos sektoriui ar specifinei kompanijai. Įvykis „building block“ yra specifinio realaus pasaulio proceso atvaizdavimas. Įvykis gali atspindėti fizinę esybę tokia kaip identifikuojamas komponentas ar informacinė esybė tokia kaip specifinis užsakymas. Pavyzdžiui gamybos procese (administracinėje aplinkoje) bus atvejų kai „building block“ nurodys skirtingus realaus pasaulio objektus. Jie yra atskiriami suteikiant jiems unikalų identitetą tačiau kitais aspektais jie yra vienodi aprašymo lygmenyje tame pačiame „building block“ kurio konkretūs atvejai jie yra.(pvz.: turbinos mentė 1 numeris 1234 ir 1235). „building block“ įvykiai gali atvaizduoti fizines ar informacines esybes. (pvz.: turbinos mentė 1234 ar užsakymo numeris 1234) (Michael Petit 2002). Bendras reliacinis standarto konstrukto modelis yra atvaizduojamas sekančiame paveiksle.



Šaltinis: Lopata A., 2002, Veiklos modelių sudėties analizė

9 Pav. CEN ENV 12 204 standarto principinė modeliavimo schema

Konstruktai dažnai yra pritaikomi skirtingais požiūriais. Konstrukto vaidmuo yra įgalinti integruotų modelių kūrimą. Požiūriai yra naudojami modelio patikrinimui iš skirtingų perspektyvų priklausomai nuo modeliuotojo tikslo. Siekiant užtikrinti nuoseklumą ir pilnumą, jie nėra naudojami modeliui struktūrizuoti. Taip pat kiekvienas konstruktas yra naudojamais skirtingais detalumo lygiais kiekviename modeliavimo etape. Taigi konstruktai nėra struktūrizuojami nei pagal požiūrius, nei pagal lygius, struktūra yra parenkama pagal loginę jų pateikimo seką. Kiekvienas modeliavimo kalbos konstruktas yra pateikiamas su iš anksto apibrėžtu bendru pritaikomu šablonu. (Michael Petit 2002)

ENV 12 204 problemos:

Dabartinė CEN ENV 12 204 standarto versija turi keletą problemų pateikiamų žemiau:

- Konstruktai yra apibūdinami savo sintaksės terminais per šablonus vietoje to tikslaus konceptualaus modelio sujungiančio koncertus į vieningą tinklą.
- ENV naudoja klasifikacijos specifikavimo sąvokas tačiau yra apibrėžiamos intuityviai. Tačiau konceptualiaame modeliavime yra gerai žinoma kad šios sąvokos gali reikšti skirtingus dalykus skirtingiems žmonėms.
 - Kai kurios sąvokos nors ir išaiškintos plačiai tačiau yra dviprasmiškos ar net prieštaraujančios viena kitai. Pavyzdžiui skirtumai ir panašumai tarp sąvokų konstruktas ir „building block“ nėra pakankamai aiškūs. Sąvoka „rolė“ taip pat yra dviprasmiška sąvoka.
- Standartas neišaiškina kas yra sukurta ENV o kas ne. Pavyzdžiui ar sąvoka „building block“ yra ENV sukurtas konceptas.

Konstrukų prasmė nėra visada aiški ir dažniausia apibrėžiama intuityviais terminais. Šie dviprasmiškumai gali vesti į skirtingą terminų supratimą. Pavyzdžiui sąvoka veiklos proceso įeiga ir išeiga galėtų būti aiškesnės. Ar gali būti objekto būseną susieta su veiklos procesu kaip funkcijos kontrolinė įeiga jei gali ką tai galėtų reikšti ?

Kartais sunku nustatyti ar konceptas apibrėžtas ENV yra klasė (skirta atvaizduoti bendras tam tikrų realaus pasaulio objektų savybes) ar meta klasė (skirta atvaizduoti bendras klasės savybes) ar meta meta klasė. (Laurent Ferrier, Patrick Heymans and Michaël Petit, 2003)

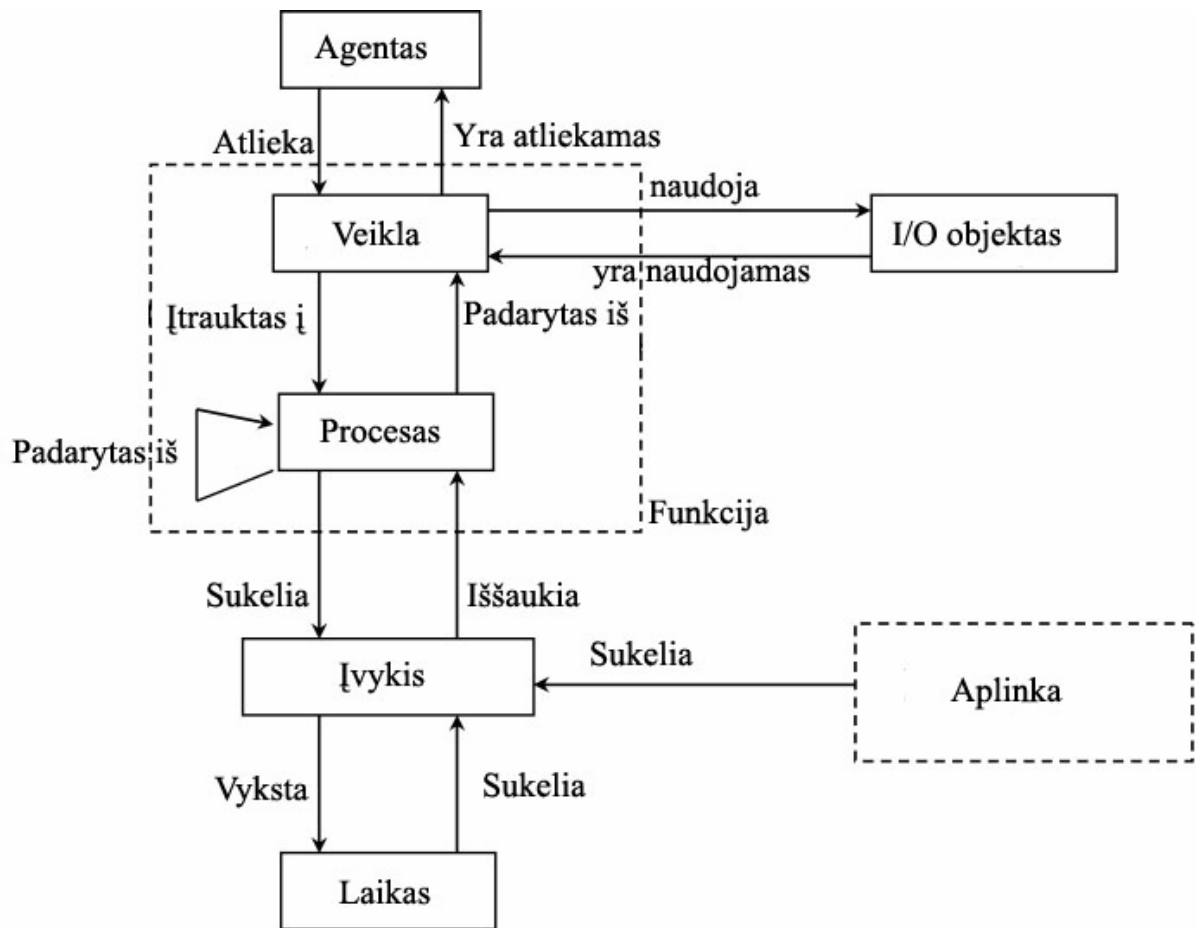
1.3.4 UEML modeliavimo kalbos analizė

Veiklos analizė ir modeliavimas yra esminiai įrankiai siekiant integralumo. Tačiau didelis skaičius skirtingų veiklos modeliavimo įrankių ir technikų esamų šiuo metu mažina veiklos modeliavimo naudingumą. Kaip šios problemos sprendimas Unified Enterprise Modelling Language (Unifikuota veiklos modeliavimo kalba) yra kuriama (IFAC-IFIP Task Force, 2001). Ši modeliavimo kalba turėtų funkcionuoti kaip tiltas tarp skirtingų modeliavimo technologijų ir įrankių, palaikydama modelių komunikavimą ir pasikeitimus (Maria Bergholtz, Paul Johannesson, Petia Wohed, 2004). UEML suteiks bendrą kalbą patenkinančią tiek techninius tiek politinius galinio vartotojo poreikius. UEML sudaro standartinis vartotojo interfeisas veiklos modeliavimui analizei ir simuliacijai. Kitas UEML tikslas yra suteikti neutralią kalbą universaliam modelių apikeitimui tarp skirtingų veiklos modeliavimo įrankių taip pat ir tarp verslo vartotojų (David Shorter, 2004).

Pagrindiniai UEML konstruktai pateikiami žemiau:

UEML branduolys = {Objektai (Įvykis, Laikas, Agentas, Procesas, Veikla, Funkcija, Įeigos objektas, Išeigos objektas, Aplinka), Ryšiai (Sukelia, Įjungia, Pagamina, Įvyksta, Sudarytas iš, Atlieka, Naudojamas);

Proceso ir funkcijos konceptų tobulinimas ir apibrėžimas yra vieni pagrindinių veiklos modeliavimo aspektų. UEML branduolys funkciją ir procesą apibrėžia kaip skirtingus Veiklos modelio konstruktus, tačiau veiklos, proceso ir funkcijos komunikavimas nėra apibrėžtas. Žemiau esančiame paveiksle matome jog konstrukto Funkcija sudėtyje yra konstruktas Procesas ir kad konstruktas Procesas yra sudarytas iš Veiklų rinkinio. Informaciniai srutai tarp Veiklos, Proceso ir Funkcijos taip pat nėra apibrėžti UEML branduolyje (Gudas S., Lopata A., Skersys T., 2002).



Šaltinis: Lopata A., 2002, Veiklos modelių sudėties analizė

10 Pav. UEML metodologijos principinė modeliavimo schema

Yra išleistas naujas unifikuotos veiklos modeliavimo kalbos leidimas UEML 1.0. Naujajame leidime pateikiamas platesnis konstruktų spektras. Formalizuotas UEML 1.0 aprašymas yra pateikiamas žemiau:

UEML_modelis={UEML_Objektas (UEML_modelis, Objektai(Informacinis objektas, Ištekliai (Žmogiškieji ištekliai, Materialūs ištekliai)), Jungtys(Išteklių rolė, Inkaras(ĮeigosJungtis, Išeigos jungtis, Jungties operatorius), Geometrija)

Srautai (I/O srautas, Išteklių srautas, Kontrolinis srautas(Trigerios srautas, apribojimo srautas)), Veikla (Įeigos jungtis, Išeigos jungtis))

Ryšiai(Pirmumo ryšys, turiIP, turiOP, Apribojimai, nešamaI/O, nešamas_Objektas, nešamas_Resuras)} (Gudas S., Lopata A., Skersys T., 2002).

1.3.5 Gudas, Lopata Veiklos metamodelio analizė

Reikia pastebėti jog UEML 1.0 versijoje yra tik du pagrindiniai į veiklos proceso modeliavimą orientuoti konstruktai: Veikla ir Srautas. UEML 1.0 versijos branduolyje nėra proceso ir funkcijos konstruktų. Galime tik spėti jog šie veiklos modeliavimo konstruktai yra paslėpti konstrukte Veikla. Konstruktas Įvykis taip yra praleistas UEML 1,0 branduolyje, nors jis buvo išskirtas ankstesnėje UEML versijoje.

Taip pat nėra aišku kurie UEML konstruktai palaiko sprendimo priėmimo mechanizmą, jei toks mechanizmas iš viso yra UEML. Nors galima teigti kad sprendimo priėmimo mechanizmas turėtų būti modeliuojamas dėl kontrolinio srauto konstrukto –sprendimo priėmimo išeigos –kuris yra aprašomas UEML 1.0 branduolyje

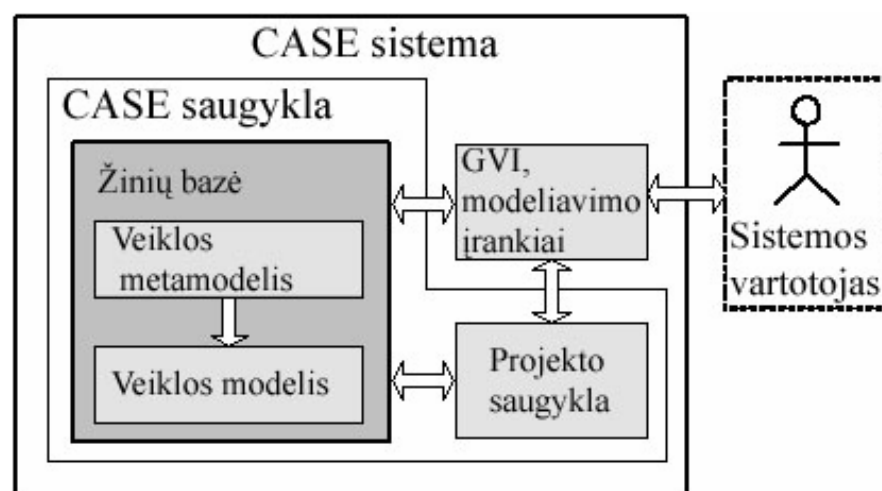
Abi UEML versijos yra susideda tik iš dviejų tipų modeliavimo konstrukčių –objektų ar esybių ir ryšių tarp objektų. Todėl tik statinė veiklos struktūra yra atvaizduojama. Dinaminiai veiklos modeliavimo aspektai nėra atvaizduojami. Taip pat ir veiklos konstrukčių transakcijų seka nėra įtraukta į UEML architektūrą.

Veiklos metamodelio vystymo procesas yra kontroliuojamas valdymo požiūriu. Veiklos metamodelis apjungia bendrus veiklos modeliavimo konstruktus iš Veiklos modeliavimo standartų ir metodologijų su naujais veiklos modeliavimo konstruktais išvystytais valdymo požiūriu.

Veiklos metamodelis yra laikomas pagrindine formalia struktūra dalykinės srities žinių rinkimui IS vystymo tikslu. Veiklos metamodelis yra sukurtas modeliais paremtam informacijos rinkimui apie probleminę sritį, rezultate gaunamas veiklos modelis apie specifinę veiklos sritį. Šis veiklos metamodeliu paremtas veiklos modelis turėtų būti dalykinės srities veiklos žinių šaltinis visuose IS kūrimo gyvavimo ciklo etapuose.

Žemiau esančiame paveiksle pavaizduota CASE sistemos architektūra su veiklos žinių posistemiū. Veiklos žinių posistemis CASE sistemoje susideda iš dviejų dalių: veiklos metamodelio (VMM) ir veiklos modelio (VM). Veiklos metamodelis yra abstraktaus lygio modelis, veiklos modelyje yra ir dalinai abstraktaus ir konkretaus lygio modelius.

Veiklos žinių posistemis turėtų būti trečiasis aktyvus informacinių sistemų kūrimo dalykinės srities žinių šaltinis (šalia analitiko ir vartotojo). Veikos metamodelis yra iš anksto nustatytų žinių šaltinis informacinių sistemų kūrimo procese. VMM naudojamas dalykinės srities žinių kaupimo procesui kontroliuoti. Taip pat VMM kontroliuoja specifinės srities veiklos modelio kūrimo procesą.



11 Pav. CASE įrankio struktūra su žinių baze

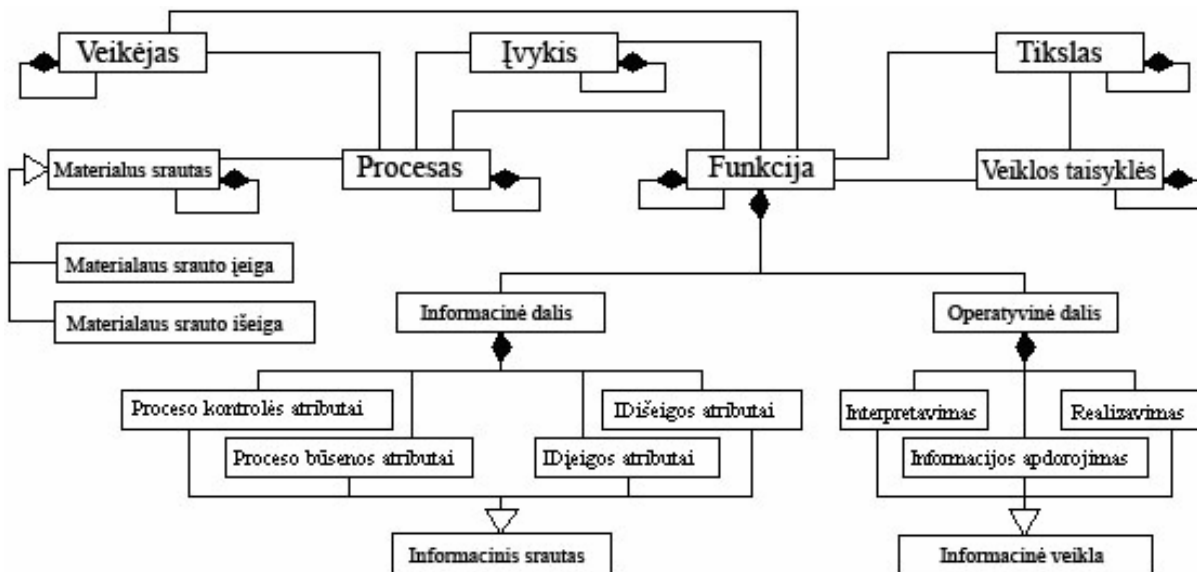
Žiniomis grindžiama IS inžinerija teigia jog visi IS kūrimo etapai yra paremti naudojamos CASE sistemos žinių baze. Case sistemos žinių bazė kartu su atitinkamais algoritmais užtikrina pastovumą tarp IS analizės ir kūrimo modelių taip pat suteikia naujas patikrinimo ir validavimo galimybes IS kūrimo ciklo etapuose. Taip pat ji gali būti naudojama veiklos procesams simuliuoti ir gerinti organizacijos viduje. Veiklos modelis kaupia dalykinės srities žinias ir yra naudojamas kaip alternatyvus žinių šaltinis IS kūrimo procese.

Case sistemos žinių bazė gali būti naudojama patikrinti dalykinės srities žinias, surinktas analitiko ir naudojamas kurti specifiniam veiklos modeliui. Tai yra atliekama patikrinant sukurtą veiklos modelį su iš anksto apibrėžta veiklos metamodelio struktūra. Verifikavimas įgalina analitiką identifikuoti loginius trūkius veiklos procesų modeliuose ir formalius trūkius kurie gali atsirasti gautuose valdymo funkcijų modeliuose.

Loginis trūkis yra semantinis trūkis tarp probleminės srities modelio (pvz.: workflow modelio) elementų. Loginės spragos atsiranda kai probleminės srities žinios yra nepilnai surenkamos.

Formalus trūkis rodo kai kurių formaliai būtinų komponentų nebuvimą tam tikrame modelyje. Šio tipo trūkiai gali būti nustatomi verifikuojant turimą modelį su formaliu veiklos metamodeliu.

Veiklos metamodelis susideda iš pagrindinių veiklos modeliavimo konceptų identifikuotų valdymo požiūriu: Veiklos procesas, funkcija, procesas, veikėjas, veiklos taisyklės, įvykis, tikslas ir su jais susiję atributai pavaizduoti žemiau esančiame paveiksle.



Šaltinis: Gudas S., Lopata A., Skersys T., 2002, Domain Knowledge Integration For Information Systems

12 Pav. Žinių bazės klasių diagrama

1.3.6 Modeliavimo kalbų, metodologijų, standartų analizės apibendrinimas

Žemiau pateikiamoje lentelėje pateikiami modeliavimo metodologijų apžvalgos konstrukto suvestinė: GERA, CIMOSA, ENV 12 204, UEML ir Gudas Lopata Veiklos modelis. Lentelėje pateikti duomenys paryškina metodologijų panašumus ir skirtumus (K.Kosanke, 1996).

1lentelė

Modeliavimo standartų metodologijų ir kalbų palyginimas

Modeliavimo konstruktai	ENV 12 204	CIMOSA	GERA	UEML	Gudas Lopata
Funkcinis požiūris	Veikla, Veiklos procesas, Įvykis	Dalykinės srities procesas; Veiklos procesas; Įvykis; Dalykinė sritis	Organizacijos funkcijos	Veikla Procesas Įvykis Funkcija	Funkcija, Įvykis, Procesas
Sprendimo priėmimo požiūris	Neapibrėžta	Neapibrėžta	Valdymas ir kontrolė konkretus požiūris	Neapibrėžta	Interpretavimas, informacijos apdorojimas, realizavimas
Organizacinis požiūris	Organizacinis vienetas	Organizacijos vienetas	Organizacijos modelis	Vykdytojas	Veiklos taisyklės, veikėjas, tikslai
Informacinis požiūris	Organizacijos objektas Rezultatas Tvarka Objekto aspektas Santykis	Organizacijos objektas Objekto požiūris Informacinis elementas Santykis	Programinė techninė įranga.	Laikas Aplinka	Informacinė veikla, informacinis srautas
Resursų požiūris	Galimybės Ištekliai	Gamybos technologijos ištekliai Informacinių technologijų ištekliai	Mašinos žmogiškieji ištekliai.	Įeigos/ išeigos objektas	Materialusis srautas materialaus srauto Įeiga/išeiga
Konstrukto skaičius	11	11	9	10	19

Šaltinis: sukurta autoriaus pagal K.Kosanke, 1996, Comparison of Modelling Methodologies ir Lopata A., 2002, Veiklos modelių sudėties analizė. Informacinės

Iš pateiktos suvestinės matome jog dauguma veiklos modeliavimo metodologijų neatsižvelgia į sprendimo priėmimo požiūrį, silpnai išvystytas organizacinis požiūris. Konstrukto gausa 19 kai tuo tarpu kitos metodologijos turi apie 10 bei pilnumu, dagingi nėra neapibrėžtų požiūrių, išsiskiria tik „Gudas Lopata Veiklos modelis“ visam IS kūrimo gyvavimo ciklui intelektualizuoti skirtas metodas. Todėl atlikus pagrindinių veiklos modeliavimo standartų, metodologijų bei modeliavimo kalbų analizę buvo nuspręsta magistro baigiamajame darbe naudoti „Gudas Lopata Veiklos modelio“ metodą kaip pagrindą sekų diagramos generavimo proceso kūrimui.

1.4. UML taikymas IS inžinerijoje

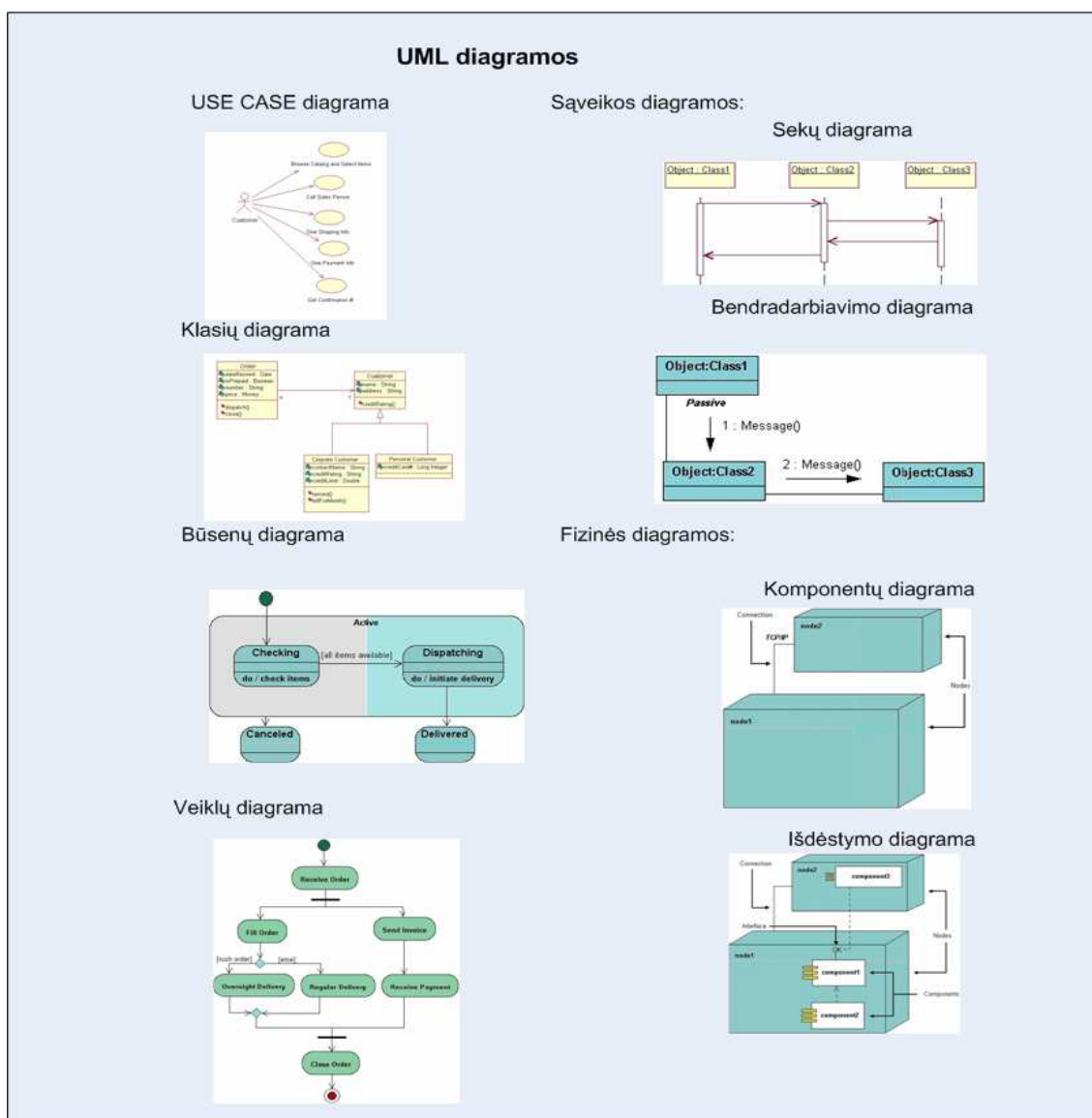
Šiame skyriuje bus trumpai aptarta UML modeliavimo kalba verslo požiūriu. UML gali būti laikomas objektiškai orientuotos analizės ir inžinerijos metodų „karų“ vykusių aštuntajame ir devintojo dešimtmečio pradžioje, nugalėtoju. UML atstovauja tik metodo kalbos komponentus ir yra atskirai papildytas RUP(angl. Rational unified process). UML atvaizduoja trijų pagrindinių modeliavimo kalbų metodų sujungimą Booch Rumbaugh ir Jacobson. UML perėjo standartizacijos procesą su Objektinio valdymo grupe (angl. Object Management Group (OMG))grupe ir dabar yra OMG standartas. UML yra modeliavimo kalba turinti savo notaciją ir semantiką (taisyklių rinkinį) kurios valdo modeliavimo kalbą. Taisyklės gali būti klasifikuojamos į:

- Sintaksinės –nustato taisyklių kryptį ir suderinimą.
- Semantinės –nustato notacijos pavienių simbolių ir kontekste reikšmes.
- Pragmatinės –nustato kalbos naudojimo gaires (notacijos tikslas)

UML architektūra susideda iš notacijos (diagramų), apribojimų ir ryšių(plėtinių) mechanizmų. UML diagramos yra trumpai apžvelgiamos sekančiuose poskyriuose (Ovidiu S. Noran, 2000).

1.4.1 UML diagramų apžvalga

UML sudaro devynios diagramos iš kurių septynios yra naudojamos veiklos modeliavimui. UML diagramų grafiniai pavyzdžiai pateikiami žemiau esančiame paveiksle.



Šaltinis: sukurta autoriaus

13 Pav. UML diagramos

The use cases diagrama aprašo sistemos funkcionalumą. Use case diagrama yra vienas tam tikro veikėjo sistemos panaudojimo scenarijus. Use case diagrama aprašo santykius tarp panaudojimo atvejų. Viena diagrama gali turėti kita, ją praplėsti ar apibendrinti.

Klasių diagrama susidedanti iš klasės ir ryšio. Aprašo sistemos struktūrą. Klasės gali atvaizduoti informaciją, produktus, dokumentus ar organizaciją.

Objektų diagrama atvaizduoja klasių diagramos atskirą atvejį specifinei situacijai.

Būsenų diagrama atvaizduoja galimas sistemos būsenas. Ši diagrama fiksuoja sistemos ir jos objektų gyvavimo ciklą, parodydama kokiose būsenos jie gali būti ir kaip skirtingi įvykiai gali paveikti būsenas bėgant laikui.

Veiklų diagrama aprašo veiklas ir veiksmus vykstančius sistemoje. Veiklos modeliavime veiklų diagramos gali būti naudojamos biznio procesams modeliuoti (paprastai apibūdinamos kaip veiklų srautai ar sekos). Toks modeliavimas atliekama per objekto ar kontrolinį srautą. Srautai gali turėti veiksmus, ar apribojimus, gali turėti siunčiamas sąlygas.

Sąveikos diagramos

Sekų diagramos aprašo žinučių tarp objektų rinkinių sekas. Laikas ir siuntimo tvarka yra aiškiai apibrėžta. Žinutės sekos diagramoje gali turėti parametrus ar sąlygas. Žinutės taip pat gali būti sinchroninės (turi būti įgyvendintos prieš pradėdant bet kokią kitą veiklą) arba asinchroninės (siuntėjas nelaukia atsakymo prieš tęsdamas kitus veiksmus)

Bendradarbiavimo diagramos yra panašios į sekų diagramas. Tačiau jos gali atvaizduoti sudėtingesnes sąveikas tarp bendradarbiaujančių objektų. Bet tai padaro jas sunkiau suprantamas nei sekų.

Fizinės diagramos

Komponentų diagramos pagrinde naudojamos atvaizduoti komponentus programinės įrangos sistemose. Kol kas jos nenaudojamos veiklos modeliavime. Išdėstymo diagramos yra klasių diagramos atvaizduojančios techninę įrangą įeinančią į programinę įrangą. Todėl savo dabartinėje formoje (UML vis dar tobulinama) jos nenaudojamos veiklos modeliavime.

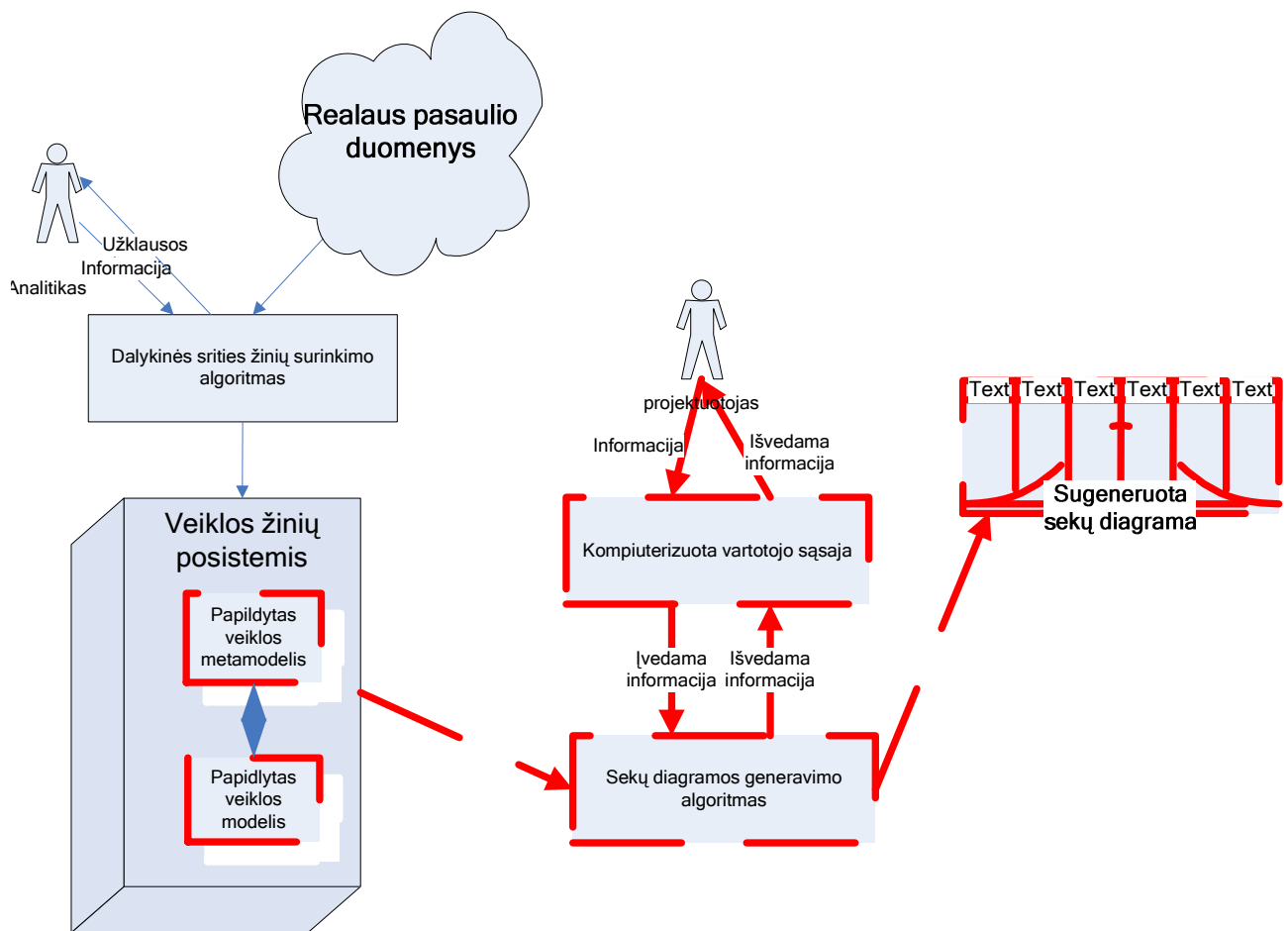
Pagrindinis UML tikslas yra užfiksuoti tris pagrindinius sistemos aspektus naudojant aukščiau paminėtas diagramas: sistemos struktūrą, elgesį ir funkcionalumą. (Ovidiu S. Noran, 2000)

1.4.2 Sekų diagramos metamodelis

Norint atlikti sekų modelio generavimą iš veiklos modelio reikia patikrinti ar dalykinės srities žinių saugykloje veiklos modelyje saugomų žinių pakanka sekų modelio generavimui. Tuo tikslu reikia sukurti sekų metamodelį atvaizduojantį sekų modelio struktūrą konceptų lygyje, jo su struktūrą išanalizuoti veiklos metamodelio aspektu siekiant išaiškinti ar veiklos modelyje saugomų žinių pakanka sekų diagramai generuoti. Autoriaus sukurtas sekų metamodelio klasių diagrama analizuojama 6.6 skyriuje

1.4.3 Sekų diagramos generavimo VM pagrindu galimybės

Siekiant išsiaiškinti sekų diagramos generavimo galimybes naudojant veiklos žinių saugyklą t.y. veiklos modelį reikia išanalizuoti veiklos modelio ir sekų modelio struktūrą ir išsiaiškinti ar pakanka veiklos modelyje saugomų žinių sekų modelio generavimui tuo tikslu. Tai atliekama metamodelio lygyje. Dekomponavus sekų modelį į jo struktūrą apibrėžiantį sekų metamodelį atliekamas turimo veiklos metamodelio ir sekų metamodelio struktūrinių vienetų: klasių ir atributų sulyginimas, tikrinama kurie elementai atsivaizduoja tiesiogiai ir kurių atvaizdavimui reikia analitiko ar sistemos projektuotojo įsikišimo principinė proceso schema yra pateikiama žemiau esančiame paveiksle.



Šaltinis: sukurta autoriaus

14 pav. Sekų modelio generavimo iš veiklos modelio principinė schema

Kaip matome aukščiau pateiktoje schemoje Sekų diagramos generavimas vykdomas iš veiklos žinių posistemyje saugomų žinių, kurios analitiko pagalba naudojant žinių surinkimo algoritmą yra surenkamos iš realaus pasaulio duomenų, patikrinamos veiklos metamodelio sudėties atžvilgiu. Surinktos žinios vėliau naudojant šiame darbe sukurtą sekų diagramos generavimo algoritmą projektuotojo pagalba atvaizduojamos generuojant sekų diagramą. Brūkšninėmis linijomis atskirti etapai bei konstruktai analizuojami šiame darbe.

1.5 Projektinių modelių generavimo galimybės egzistuojančiais CASE įrankiais

Sekančiame skyriuje yra atliekama palyginamoji populiariausių CASE įrankių palyginamoji analizė projektinių modelių generavimo aspektu taip pat apžvelgiant ir kitas IS kūrimo etapų automatizavimo galimybes.

1.5.1 Oracle developer suite 10g

“Oracle developer suite 10g” automatizuotame IS kūrimo įrankyje sistemos kūrimas prasideda dalykinės srities žinių aprašymu procesų diagramoje (angl. Process Modeler). Šioje diagramoje kuriami objektai (angl. Organisational units) kurie atlieka tam tikras veiklas (angl. process steps) taip pat perneša informacinius bei materialius srautus (angl. flows), kurie gali būti saugomi informacinėse saugyklose (angl. Store). Sukūrus procesų diagramą atliekamas interaktyvus Duomenų srautų diagramos generavimas (angl. dataflow diagrammer) t.y. diagramos yra generuojamos sukurtiems procesams su galimybe jas papildyti trūkstamais elementais. Toliau generuojama Funkcijų hierarchijos diagrama (angl. function hierarchy diagrammer) iš jau prieš tai sukurtų funkcijų, į ją įtraukiamos visos galimos pateikiamos funkcijos. Vėliau šių funkcijų pagalba generuojami moduliai. Sekančiame IS kūrimo etape yra kuriama esybių ryšių diagrama (angl. ER-entity relationship) kurioje aprašomi objektai jų ryšiai bei atributai reikalingi IS veiklos realizavimui. Pagal sukurtą ER diagramą yra generuojama duomenų bazė, formos užklauskos meniu ir ataskaitos naudojant duomenų bazės generavimo modulį(angl. database design transformer) bei aplikacijos generavimo modulį (angl. Application design transformer). Prieš kuriant galutinį IS variantą surinktos žinios yra verifikuojamos generuojant serverio modelį (angl. Server model) (Saonee Sarker, 2000).

1.5.2 Provision Workbench v4.0

Proforma Corp. Provision Workbench automatizuotame IS kūrimo įrankyje pradinės dalykinės srities žinios yra surenkamos Veiklos sąveikų modelyje (angl. Business interaction model). Šiame modelyje surenkami duomenys apie organizaciją bei jos aplinką konkurentus klientus bei tiekėjus. Sekančiame žingsnyje veiklos sąveikų modelio pagrindu dalinai generuojamas yra organizacijos modelis (angl. Organisation model) atvaizduojantis veikloje dalyvaujančius padalinius ir jų hierarchinę struktūrą. Tuo pačiu principu generuojami yra darbų sekų ir procesų modeliai. Kaip papildomi projektinio IS kūrimo etapo modeliai gali būti kuriami USE CASE, tikslų, vietovių, galimybių bei kiti modeliai. Sekančiame etape duomenų bazių struktūroms reikalingi modeliai yra aprašomi klasių diagramoje kurios pagrindu kuriamos yra subtipų diagrama, būsenų diagrama aprašanti panaudojimo scenarijus, sekų diagrama aprašanti organizacijos funkcijas

realiu laiku. Sukurtos diagramos gali būti eksportuojamos į C++ programinį kodą, MS Excel failą, iš jų taip pat gali būti generuojama MS Access duomenų bazė. Taip pat Provision Workbench įrankis siūlo ataskaitų generavimą iš sukurtos MS Access duomenų bazės (Proforma corp., 2007).

1.5.3 System Architect v8.8.13

„Popkin System Architect“ automatizuotas IS kūrimo įrankis paremtas J. Zachman ISA apibendrinto modeliu. Pasak Pupkin „Zachman architektūra nėra metodas tai galvojimo būdas“. Sistemos modeliavimas CASE įrankyje pradedamas nuo procesų diagramos (angl. Process Chart) kurioje apibrėžiami pagrindiniai organizacijos procesai informaciniai, materialūs srautai bei žmogiškieji resursai reikalingi organizacijos darbui vykdyti. Procesų diagrama gali būti dekomponuojama detalizuotoje procesų diagramoje (angl. Process thread diagram). Toliau yra apibrėžiamos pagrindinės organizacijos funkcijos funkcijų hierarchijos diagramoje (angl. Function hierarchy diagram) procesų santykis su funkcijomis yra atvaizduojamas procesų funkcijų matricoje kuri yra generuojama automatiškai. System Architect įrankis palaiko USE CASE paremtą UML modeliavimą t.y. vartotojo poreikiais paremtą modeliavimą. Iš pradžių yra kuriami UCM visiems sistemos veikimo esamiems ir norimiems scenarijams. UCM yra identifikuojami objektai, kurių pagrindų yra kuriamos bendradarbiavimo (angl. Collaboration) ir sekų (angl. Sequence) diagramos aprašančios sistemos dinaminę pusę. Iš bendradarbiavimo diagramos CASE įrankis gali automatiškai sugeneruoti sekų diagramą ir atvirkščiai. Sekančiame etape dalinai sugeneruojama klasių diagrama (angl. Class diagram) aprašanti sistemos statinę dalį t.y. sistemos objektus jų atributus bei ryšius tarp jų. Iš kurios generuojamas programinis kodas į JAVA, C++, KOBRA ir kitas programavimo kalbas. Klasių diagrama automatiškai gali būti konvertuojama į esybių ryšių diagrama (angl. ER diagram) System Architect pagalba.. Pagal ER diagramą generuojamos ORACLE, Access ar kitos DBVS loginės schemas. Šis CASE įrankis palaiko atgalinę inžineriją (angl. Reverse Engineering) t.y. modelių generavimą atgal iš kodo siekiant įvesti pataisymus. Kaip papildomos diagramos IS projektavimui gali būti kuriamos būsenų diagrama (angl. State diagram), komponentų diagrama (angl. Component diagram) ir išdėstymo diagrama (angl. Deployment diagram). Taip pat System architect pagalba galima generuoti ataskaitas ir kitą dokumentaciją reikalingą pilnam projekto įgyvendinimui (Telecogic, 2007).

1.5.4 Magic Draw UML 7.1

Magic Draw CASE įrankyje IS kūrimo ciklas atvaizduojamas penkiais požiūriais: Vartotojo požiūriu (angl. USE CASE view), loginiu požiūriu (angl. Logical view), įgyvendinimo požiūriu (angl. Implementation view), procesų požiūriu (angl. Process view) ir išdėstymo požiūriu (angl. Deployment view). Kiekvienas iš šių požiūrių aprašomas atitinkamomis UML diagramomis. Sistemos projektavimas pradedamas dalykinės srities žinių surinkimu USE CASE diagramoje.

Sistemos dinamika aprašoma sąveikų diagramomis t.y. bendradarbiavimo ir sekų, taip pat būsenų diagrama. Projektuojamos programinės įrangos struktūra atvaizduojama klasių diagramoje. Programinės fizinis išdėstymas bei įgyvendinimo problematika atvaizduojama komponenčių bei išdėstymo diagramose. Iš klasių diagramos gali būti generuojamas programinis kodas į JAVA, C++ bei kitas programavimo kalbas, taip pat gali būti generuojami DLL failai. Įrankis palaiko atgalinę inžineriją taip pat yra modelių bei projektų eksportavimo galimybes į kitus CASE įrankius tokius kaip Rational Rose. Įrankis turi plačias ataskaitų generavimo galimybes projektiniams modeliams į HTML, XML ar PDF formatus.(No Magic, Inc, 2007)

1.5.6 CASE įrankių palyginimas

2 lentelėje yra pateikiamas populiariausių šiuo metu CASE įrankių analizės apibendrinimas
2lentelė

Populiariausių CASE įrankių palyginimas

Požymis/Case įrankis	Oracle developer suite 10g	Provision Workbench v4.0	System architect v8.8.13	Magic Draw UML 7.1
Pradinė dalykinės srities žinių saugykla	Procesų modelis	Veiklos sąveikų modelis(BIM), Veiklos klasių modelis	Procesų diagrama, USE CASE modelis	Use Case modelis
Veiklos modelis	-	+	-	-
Projektinių modelių generavimo galimybės				
IS projektuotojo kuriami modeliai	Procesų modelis, ER diagrama	Veiklos sąveikų modelis, Tikslų modelis, Vietovių modelis, Klasių modelis, Organizacijos modelis	Procesų diagrama, Funkcijų hierarchijos diagrama, UCM, Sekų diagrama, Komponentų diagrama, Išdėstymo diagrama	UCM, Klasių diagrama, Sekų diagrama, Bendradarbiavimo diagrama, Būsenu diagrama, Išdėstymo diagrama, komponentų diagrama, Veiklų diagrama
Dalinai generuojami modeliai	Duomenų srautų diagrama	USE CASE modelis, Būsenu modelis, Sekų modelis, Subtipų modelis, Įvykių modelis	Detalizuota procesų diagrama, Klasių diagrama, Būsenu diagrama	-
Generuojami modeliai	Funkcijų hierarchijos diagrama, Serverio diagrama	Procesų modelis, darbų sekų modelis	Procesų funkcijų matrica, Bendradarbiavimo diagrama, ER diagrama	-
Kitų IS kūrimo etapų automatizavimo galimybės CASE įrankiuose				
Duomenų bazės loginės schemas generavimas	+	+	+	-
Kodo generavimas	-	+	+	+
Atgalinė inžinerija	-	-	+	+
DB struktūros generavimas	+	-	-	-
Dokumentacijos generavimas	-	+	+	+

Šaltinis: Sukurta autoriaus

Išanalizavus populiariausius keturis CASE įrankius projektinių modelių, kodo ir dokumentacijos generavimo aspektu galime teigti, jog prasčiausiai šiuo metu yra išvystytos modelių generavimo galimybės, dauguma programinių paketų: Oracle Developer 10g, System Architect orientuojasi į kodo, dokumentacijos generavimą bei atgalinę inžineriją. Tik Provision Workbench 4.0 turi veiklos modelį kuriame saugomos dalykinės srities žinios. Visi kiti programiniai paketai įgyvendina tik diagramų generavimą vieną iš kitos arba dalinių jų generavimą. Programa Magic Draw UML7.1 iš vis nenumato projektinių modelių kūrimo automatizavimo galimybės visi modeliai kuriami nepriklausomai vienas nuo kito. Todėl buvo nuspręsta šiame darbe kurti naują UML sekų diagramos generavimo iš veiklos modelio procesą automatizuojantį algoritmą ir programinės įrangos prototipą.

TEORINĖS DALIES IŠVADOS

1. Atlikus pagrindinių veiklos modeliavimo standartų, metodologijų bei modeliavimo kalbų analizę buvo nuspręsta magistro baigiamajame darbe naudoti Gudas Lopata veiklos modelio metodą kaip pagrindą sekų diagramos generavimo proceso kūrimui. Kadangi kitos analizuotos veiklos modeliavimo metodologijos neatsižvelgia į sprendimo priėmimo požiūrį, silpnai išvystytas organizacinis požiūris Gudas Lopata veiklos modelio metodas turi pilnai apibrėžtus visus analizuotus požiūrius. Taip pat išsiskiria modeliavimo konstruktyvų skaičiumi 19 kai tuo tarpu kitos metodologijos turi apie 10.

2. Išanalizavus Žiniomis grindžiamos inžinerijos IV etapo pagrindus bei nustatius veiklos modelio vietą joje buvo nustatyta jog norint sugeneruoti UML sekų diagramą reikia sukurti sekų diagramos generavimo algoritmą bei programinės įrangos paketą su grafine vartotojo sąsaja įgyvendinančia minėtą algoritmą.

2. Išanalizavus populiariausių šiuo metu CASE įrankių analizė buvo nustatyta jog jų projektinių modelių generavimo galimybės yra nepakankamos, nes dauguma programinių paketų: Oracle Developer 10g, System Architect orientuojasi į kodo, dokumentacijos generavimą bei atgalinę inžineriją. Tik Provision Workbench 4.0 turi veiklos modelį kuriame saugomos dalykinės srities žinios. Visi kiti programiniai paketai įgyvendina tik diagramų generavimą vieną iš kitos arba dalinį jų generavimą. Programa Magic Draw UML7.1 iš vis nenumato projektinių modelių kūrimo automatizavimo galimybės visi modeliai kuriami nepriklausomai vienas nuo kito. Todėl buvo nuspręsta šiame darbe kurti naują UML sekų diagramos generavimo iš veiklos modelio procesą automatizuojantį algoritmą ir jį pagrindžiantį programinės įrangos prototipą.

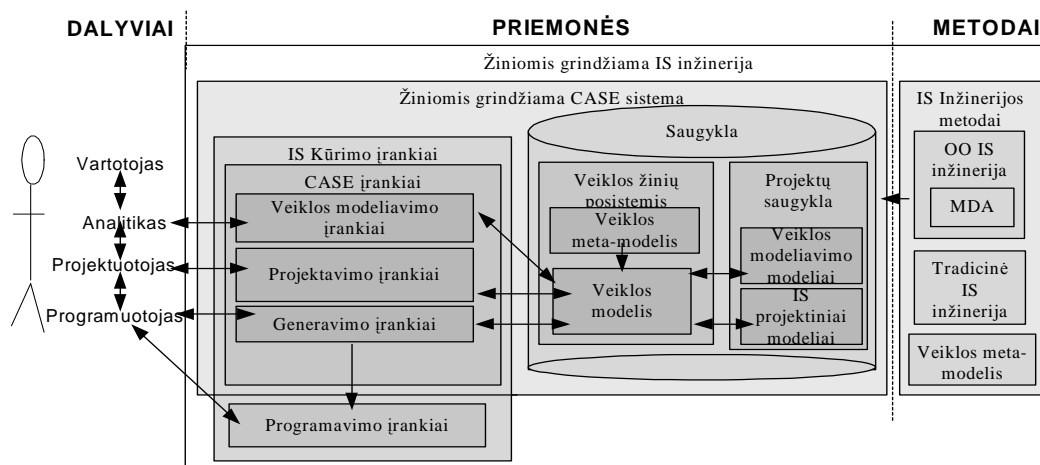
2. VEIKLOS MODELIU GRINDŽIAMO UML SEKŲ MODELIO GENERAVIMO GALIMYBĖS ANALIZĖ

2.1 Žiniomis grindžiamos IS inžinerijos IV etapo analizė

Ketvirtajame IS inžinerijos vystymosi etape, kuriame integruojasi trijų nagrinėtų sričių (IS inžinerijos, veiklos modeliavimo, dirbtinio intelekto) rezultatai, CASE sistemos dalimi turi tapti veiklos žinių posistemis. Veiklos žinių saugykloje saugomos visos IS projektinių modelių kaupimui būtinos žinios apie veiklos sritį (organizacijos procesus ir funkcijas) (Gudas S., Lopata A. 2004). Žinių bazė tai saugykla, kurioje saugomą informaciją apibrėžia metaduomenys. Veiklos žinių saugyklos sudėtį apibrėžia formalizuota žinių apie veiklos valdymą struktūra – veiklos metamodelis (Gudas, Skersys, Lopata, 2004). Žiniomis grindžiama IS inžinerija apibrėžiama kaip inžinerijos metodologija, kuri reglamentuoja žinių, būtinų produktui projektuoti, analizuoti ir kurti saugojimą specialiaame produkto modelyje (Gudas S., Lopata A. 2005).

Šio etapo ypatumas – CASE sistemose integruojasi kompiuterizuotų IS inžinerijos metodų kūrimo srities (metodų inžinerija), veiklos modeliavimo, organizacijų projektavimo (veiklos reinžinerijos) ir žinių inžinerijos (dirbtinio intelekto, organizacijos žinių) sričių metodai ir modeliai.

Veiklos žinių kaupimo posistemį sudaro du pagrindiniai elementai: veiklos modelis ir veiklos metamodelis. Veiklos metamodelis – tai formali žinių struktūra, apibrėžianti veiklos srities esminius komponentų tipus ir jų sąveikų tipus. Žiniomis grindžiama CASE įrankio žinių bazėje kaupiamos dalykinės srities žinios yra patikrintos veiklos metamodelio atžvilgiu. Veiklos žinių posistemio paskirtis teoriškai yra užtikrinti galimybę saugomų žinių pagrindu generuoti IS konceptualaus ir detalaus projektavimo etapų modelius ir programinį kodą (Gudas S., Lopata A. 2004). Ketvirto IS inžinerijos vystymosi etapo principinė schema pateikta 15 paveiksle.



Šaltinis: Gudas S., Lopata A. (2005) Žiniomis grindžiamos informacijos sistemų inžinerijos bruožai

15 pav. Ketvirto IS inžinerijos vystymosi etapo principinė schema

IS inžinerijos ketvirtojo raidos etapo principinės schemos apibūdinimas:

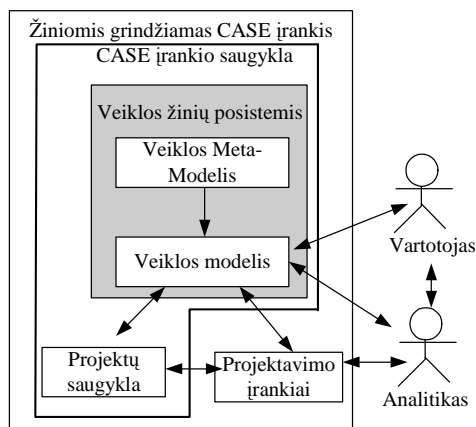
- Dalyviai: vartotojas, analitikas, projektuotojas, programuotojas, CASE sistemos veiklos žinių posistemis;
- Metodai: žiniomis grindžiami CASE metodai;
- Modeliai: a) IS projekto modeliai: struktūriniai–funkciniai, objektiškai orientuoti arba modifikuoti CASE priemonės gamintojų metodai; b) Veiklos modeliai; c) Veiklos žinių bazė, sudaryta veiklos meta–modelio pagrindu;
- Priemonės – žiniomis grindžiamos CASE sistemos su veiklos žinių posistemiu, kurio šerdis – veiklos žinių bazė, sudaryta veiklos meta–modelio pagrindu (Gudas S., Lopata A. 2004).

2.2 Veiklos žinių kaupimo posistemio analizė

Veiklos žinių posistemis išplečia šiuo metu egzistuojančių CASE įrankių architektūrą atlikdamas pagrindines kompiuterizuojamos dalykinės srities žinių saugyklos funkcijas. IS projektavimo etape veiklos modelis, kaip pagrindinė organizacijos žinių struktūra, kontroliuoja IS projektavimo eigą, sumažindamas IS projektavimo loginių klaidų ir trūkių tikimybę.

Tradicinio CASE įrankio saugykla papildoma veiklos žinių kaupimo posistemiu, kurį sudaro veiklos modelis ir jo sudarymo tvarką reglamentuojantis veiklos metamodelis. Visi projektavimo įrankiais sukurti projektiniai modeliai, sudaryti veiklos modelyje sukauptų žinių pagrindu, saugomi projektų saugykloje. Veiklos modelį konkrečiai kompiuterizuojamai dalykinei veiklos sričiai, atsižvelgdami į veiklos metamodelio apribojimus, reikiama informacija užpildo vartotojas ir analitikas (Gudas S., Skersys T., Lopata A. 2004).

Veiklos modelio vieta žiniomis grindžiamame CASE įrankyje pavaizduota 16 paveiksle. Toliau darbe analizuojamas veiklos metamodelis, sekų metamodelis, bei sekų modelio generavimo galimybės.



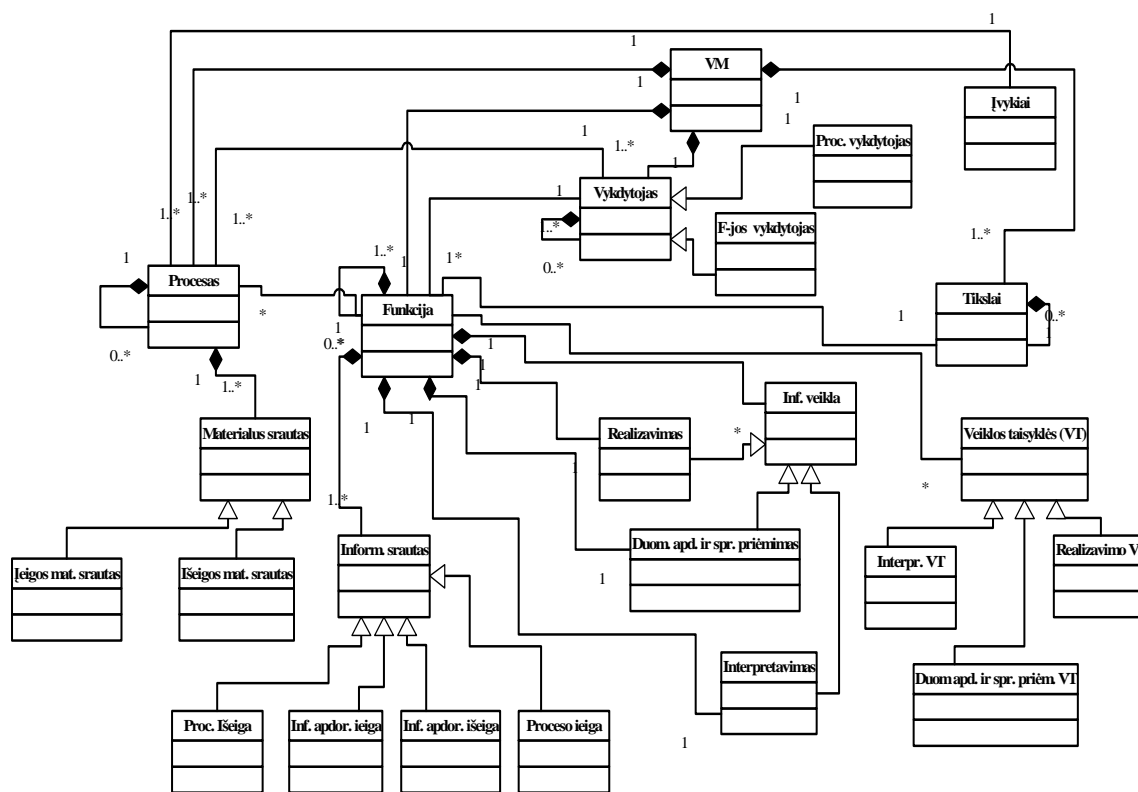
Šaltinis: Gudas S., Lopata A, 2005, Žiniomis grindžiamos IS inžinerijos bruožai

16 pav. veiklos modelio vieta žiniomis grindžiamame CASE įrankyje

2.3 Veiklos metamodelio analizė

Veiklos metamodelis – formali bazinė dalykinės srities žinių struktūra (Gudas S., Skersys T., Lopata A. 2004. p7)

Veiklos modelio pagalba, yra modeliuojami įmonėje vykstantys procesai, įvertinant aplinkos poveikį organizacijai. Veiklos modeliavimo sąvoka apima šias sudėtingas dalis: funkcijas, elgseną, išteklius ir organizaciją. Reikia pabrėžti, kad informacinių sistemų inžinerijoje labai svarbu unifikuoti veiklos modeliavimą, kadangi dabar funkcionuojantys veiklos modeliai neatitinka veiklos modeliavimo reikalavimų informacinių sistemų inžinerijos srityje. Todėl IS inžinerijoje yra siekiama naudoti formalizuotą žinių apie veiklos valdymą struktūrą – veiklos metamodelį. Veiklos metamodelyje yra integruoti skirtingi veiklos modeliavimo požiūriai ir metodikos. Yra išskiriami du esminiai veiklos meta modelio elementai: funkcija ir procesas. Procesas apibūdina materialią, o funkcija – informacinę veiklos pusę.

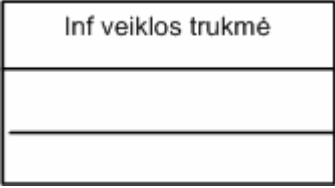
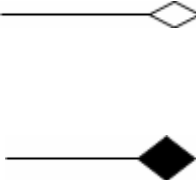
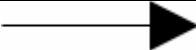


Šaltinis: Gudas S. Lopata A., 2005, Veiklos modeliu grindžiamas kompiuterizuotas funkcinų vartotojo reikalavimų specifikavimo metodas

17 pav. Veiklos metamodelio klasių modelis

Metamodelių klasių modelio notacijos aprašymas pateikiamas žemiau:

Veiklos metamodelio sudarymo notacija

Eil. Nr.	Žymėjimas	Pavadinimas	Apibūdinimas
1.		Klasė	Apibūdina kuriamo metamodelio konstruktus
2.		Sudėtinė dalis	Nurodo jog vienas objektas yra kito sudedamoji dalis užtušotas rombas reiškia neprivalomą sudėtinį konstruktą, tuščiaaviduris rombas reiškia, jog elementas yra privaloma sudėtinė dalis.
3.		Tipas	Šis žymėjimas atvaizduoja metamodelio objektų tipus.
4.	1..1,1..*	Kardinalumas	Šis žymėjimas nurodo ryšio tarp metamodelio objektų kardinalumą.

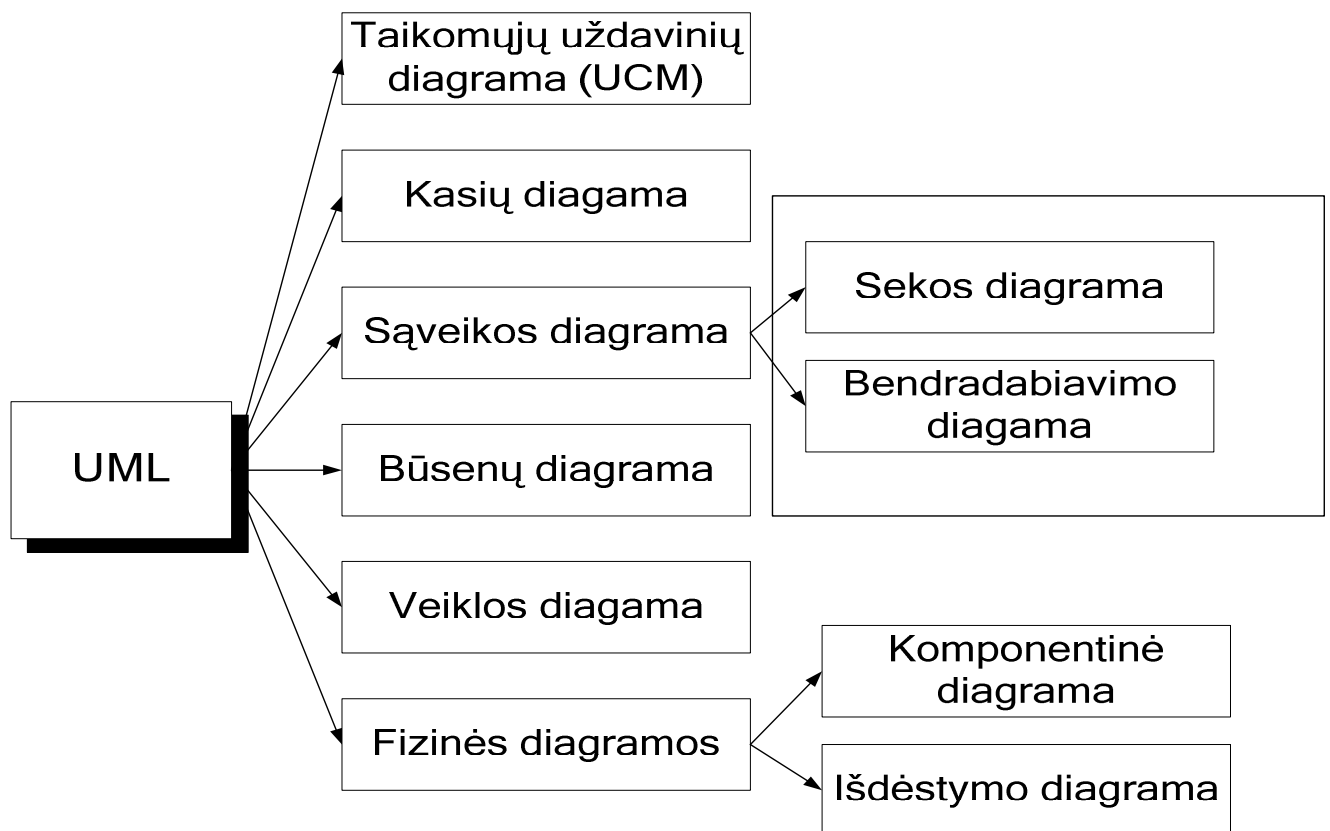
Šaltinis: sukurta autoriaus

Veiklos metamodelyje (žr. 17 pav.) galima išskirti šiuos pagrindinius veiklos metamodelio konstrukcinius elementus:

Veiklos **Procesas į kurį įeina Materialus srautas galintis būti sekančių tipų:** Įeigos materialusis srautas, Išeigos materialusis srautas. Veiklos funkcijos kurios atributas yra informacinis srautas galintis būti keturių tipų: Proceso įeiga, Informacijos apdorojimo įeiga, Informacijos apdorojimo išeiga, Proceso išeiga. Taip pat funkcijos elementai yra Interpretavimas, Duomenų apdorojimas ir sprendimų priėmimas, Realizavimas. Šie elementai savo ruožtu yra informacinės veiklos atributai, kuri taip pat yra funkcijos sudedamoji dalis. Taip pat visi procesai ir funkcijos turi turėti **Vykdytoją kuris gali būti dviejų tipų:** Proceso vykdytojas, Funkcijos vykdytojas. Visa organizacijos veikla remiasi tikslais ir veiklos taisyklėmis kurios gali būti sekančių tipų: Interpretavimo veiklos taisyklės, realizavimo veiklos taisyklės, Duomenų apdorojimo ir sprendimo priėmimo Veiklos taisyklės. Procesus veiklos modelyje inicijuoja įvykiai. Veiklos procesai, funkcijos vykdytojai bei tikslai tarpusavyje yra susiję kaip parodyta 17 paveiksle.

2.4 Analizuojami UML modeliai

UML (ang. Unified Modeling Language) – universali modeliavimo kalba, nurodanti standartinių rinkinių diagramų ir notacijų, skirtų objektiškai orientuotų sistemų modeliavimui, aprašanti diagramų ir simbolių semantines reikšmes, skirta programinei įrangai specifikuoti, konstruoti, vizualizuoti ir dokumentuoti (Object Management Group, 2003, p. 500-502). 18 paveiksle yra pavaizduotos UML diagramos skirstomos į dvi pagrindines rūšis: statines ir dinamines. Statinę sistemos dalį atvaizduoja šios diagramos: klasių, fizinės diagramos kurios savo ruožtu skirstomos į: komponentinės ir išsidėstymo, būsenų diagramos. Sistemos dinamikos modeliavimas atliekamas taikant šias UML diagramas: Taikomųjų uždavinių, veiklos, sąveikos diagramos kurių sudaro: sekos ir bendradarbiavimo diagramos.



Šaltinis: sukurta autoriaus

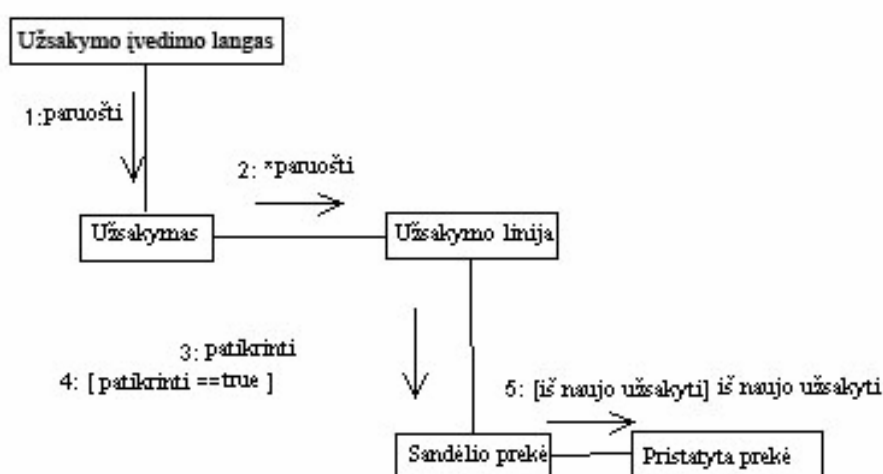
18 pav. Analizuojamos UML diagramos

Šiame darbe analizuojamos sistemos dinaminę pusę atvaizduojančios sekų diagramos generavimo veiklos modelio pagrindu galimybės, kuriamas algoritmas ir programinės įrangos realizuojančios jį prototipas.

2.5 UML sąveikos modelių analizė

Į objektų tarpusavio sąveikos sampratą yra įtraukiami du aspektai: objektų tarpusavio komunikacija ir objektų tarpusavio ryšiai. Atitinkamai sąveikos proceso elementams galima išskirti dviejų tipų sąveikos modelius, kuriuos galima sukurti panaudojant UML modeliavimo kalbą: (Miller R. 2003)

1. **Bendradarbiavimo(angl. collaboration).** Šioje diagramoje komunikaciniai ryšiai nėra pateikiami realiame laike, todėl komunikacijos eiga yra numeruojama (Nuno Jardim Nunes, João Falcão e Cunha, 2000).

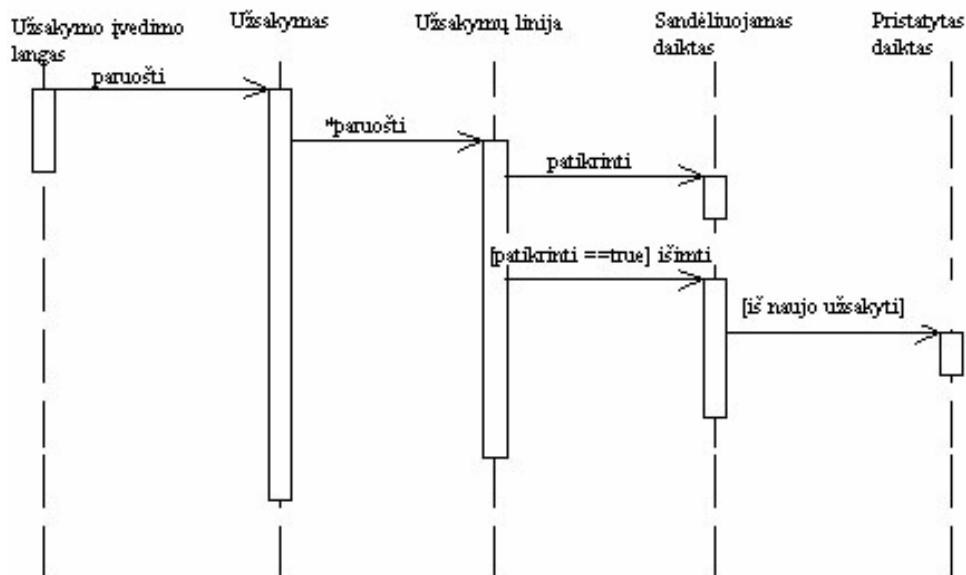


Šaltinis: Nuno Jardim Nunes, João Falcão e Cunha, 2000, Towards a UML profile for interaction design: the Wisdom approach

19 pav. Bendradarbiavimo diagramos pavyzdys

Nagrinėjant detaliau bendradarbiavimo diagramos (žr. 19 pav.) ypatumus, galima pastebėti, kad bendradarbiavimo diagramoje objektai yra pavaizduojami ikonomis, o informacijos pateikimo seka bendradarbiavimo diagramoje yra numeruojama. UML yra naudojama numeracija sudarant bendradarbiavimo diagramą, siekiant aiškiau parodyti operacijų tarpusavio sąsajas (Larman C. 2005).

2. **Sekos (angl. sequence).** Ši diagrama yra naudinga, nagrinėjant komunikacijos proceso eigą realiu laiku, analizuojant sudėtingus scenarijus. Sekų diagrama dekomponuoja organizacijoje vykstančio komunikacinio proceso eigą (Larman C. 2005). Sekos diagrama turi dvi dimensijas: horizontaliąją ir vertikaliąją (žr. 20 pav.). Seka realiame laike yra atvaizduojama iš viršaus žemyn. Galima pastebėti, kad jeigu norima dimensijos gali būti sukeistos (Louisiana State University 2005). Be to, sekos diagramose svarbu yra pats laiko sekos atvaizdavimas, tačiau įvykiams vykstant realiame laike, yra naudojama ir laiko matavimo skalė.



Šaltinis: Louisiana State University (2005) Interaction diagrams

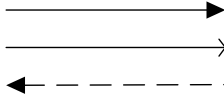

20 pav. Sekos diagramos pavyzdys

Sekų diagramos notacijos aprašymas yra pateikiamas žemiau esančioje lentelėje.

4lentelė

UML sekų diagramos notacija

Eil. Nr.	Apibūdinimas	Pavadinimas	Žymėjimas
1.	Kiekvienas objekto gyvavimo ciklas atvaizduojamas sekų diagramos viršuje iš objekto žemyn brėžiama brūkšninė linija nurodanti objekto gyvavimą laiko aspektu. Rodyklės pasibaigiančios ant objekto gyvavimo ciklo linijos rodo kad pranešimas į objektą ateina, išeinančios iš jo rodo, kad pranešimas iš objekto yra siunčiamas. Laikas sekų diagramoje eina iš viršaus į apačią vertikalčiai.	Objekto gyvavimo ciklas	
2.	Sekų diagramos vykdytojas yra pirmasis objektas brėžiamas sekų diagramoje dažniausiai jis pradeda pranešimų seką ir gauna paskutinį pranešimą kuriuo išvedama informacija.	Vykdytojas	
3.	Aktyvacija rodo kada objektas atlieka tam tikrus procesus ar skaičiavimus laike. Aktyvacija vaizduojama kaip plonas stačiakampis brėžiamas ant objekto gyvavimo ciklo.	Aktyvacija	

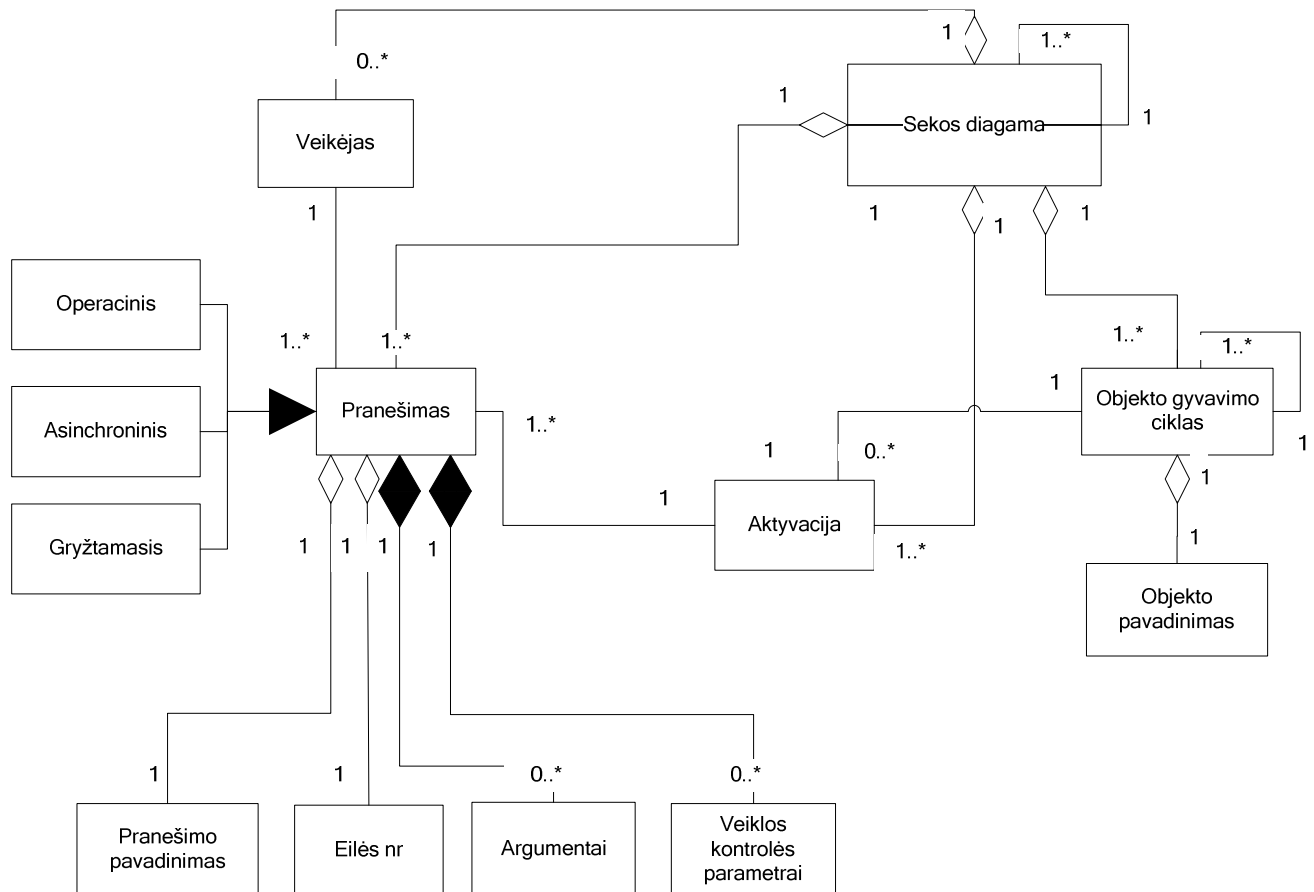
4.	<p>Pranešimas sekų diagramose atvaizduojamas kaip horizontali linija jungianti vieną objektą su kitu. Pranešimas gali perduoti ne tik tekstą bet ir funkcijas, jų parametrus, apribojimus ir kitą informaciją reikalingą organizacijos funkcijose vykstančių sekų dekompozicijai. Pranešimai gali būti trijų rūšių priklausomai nuo jų paskirties: operaciniai, asinchroniniai ir grįžtamieji. Jų atitinkamas žymėjimas matomas dešiniau esančiame stulpelyje. Operaciniai pranešimai perneša informaciją, parametrus reikalingą operacijoms atlikti. Asinchroniniai pranešimai –siunčiami kartu su operaciniais tačiau tolesniam sekos vykdymui skirtingai nei operaciniuose pranešimuose, nereikia sulaukti atsakymo seka vykdoma toliau. Grįžtamieji pranešimai gražina užklausų, funkcijų, skaičiavimų atsakymus į objektą iš kurio buvo siunčiamas operacinis pranešimas.</p>	Pranešimas	
5.	<p>Ištrynimasis indikuoja objekto gyvavimo ciklo aktyvacijos pabaigą. Ir žymimas X objekto aktyvacijos pabaigoje.</p>	Ištrynimasis	

Šaltinis: sukurta autoriaus pagal Object Management Group, OMG, 2003, Unified Modeling Language specification

Sekančiame skyriuje yra pateikiamas ir analizuojamas autoriaus sukurtas UML sekų diagramos metamodelis ir jo sudėtis.

2.6. Siūlomas UML Sekų metamodelis

21 pav. yra pateikiamas UML sekų metamodelis. Metamodelyje yra pateikiama sekų diagramos struktūriniai elementai jų tarpusavio ryšiai bei atributai.



Šaltinis: sukurta autoriaus

21 pav. Siūlomas UML sekų metamodelis

Pagrindiniai sekos metamodelio konstrukciniai elementai yra:

Objektų gyvavimo ciklas turintis viena pavadinimą, veikėjas atliekantis informacinę veiklą pranešimai siunčiami tarp objektų gyvavimo ciklų gali būti trijų tipų : Operacinis, kuriuo siunčiama pagrindinė sekos informacija, argumentai, Asinchroninis, kuriuo siunčiama papildoma informacija, Gryžtamasis ryšys grąžina operacinio pranešimo informaciją. Pranešimas privalo turėti pavadinimą bei eilės numerį, neprivalomai pranešimas gali turėti procedūrų argumentus bei veiklos kontrolės parametrus. Aktyvacija –svarbi sekų modelio dedamoji dalis įprasminanti realaus laiko įvykius diagramoje parodo kada būna aktyvus objekto gyvavimo ciklas realiu laiku..

2.7 Veiklos metamodelio atvaizdavimas į sekų metamodelį

1 lentelėje matome veiklos metamodeliu grindžiamo UML sekų metamodelio atvaizdavimo proceso formalų aprašymą susidedantį iš veiklos ir sekų metamodelių atitinkamos klasės pavadinimų ir atvaizdžio nurodančio kaip siejasi metamodelių klasės viena su kita.

5lentelė

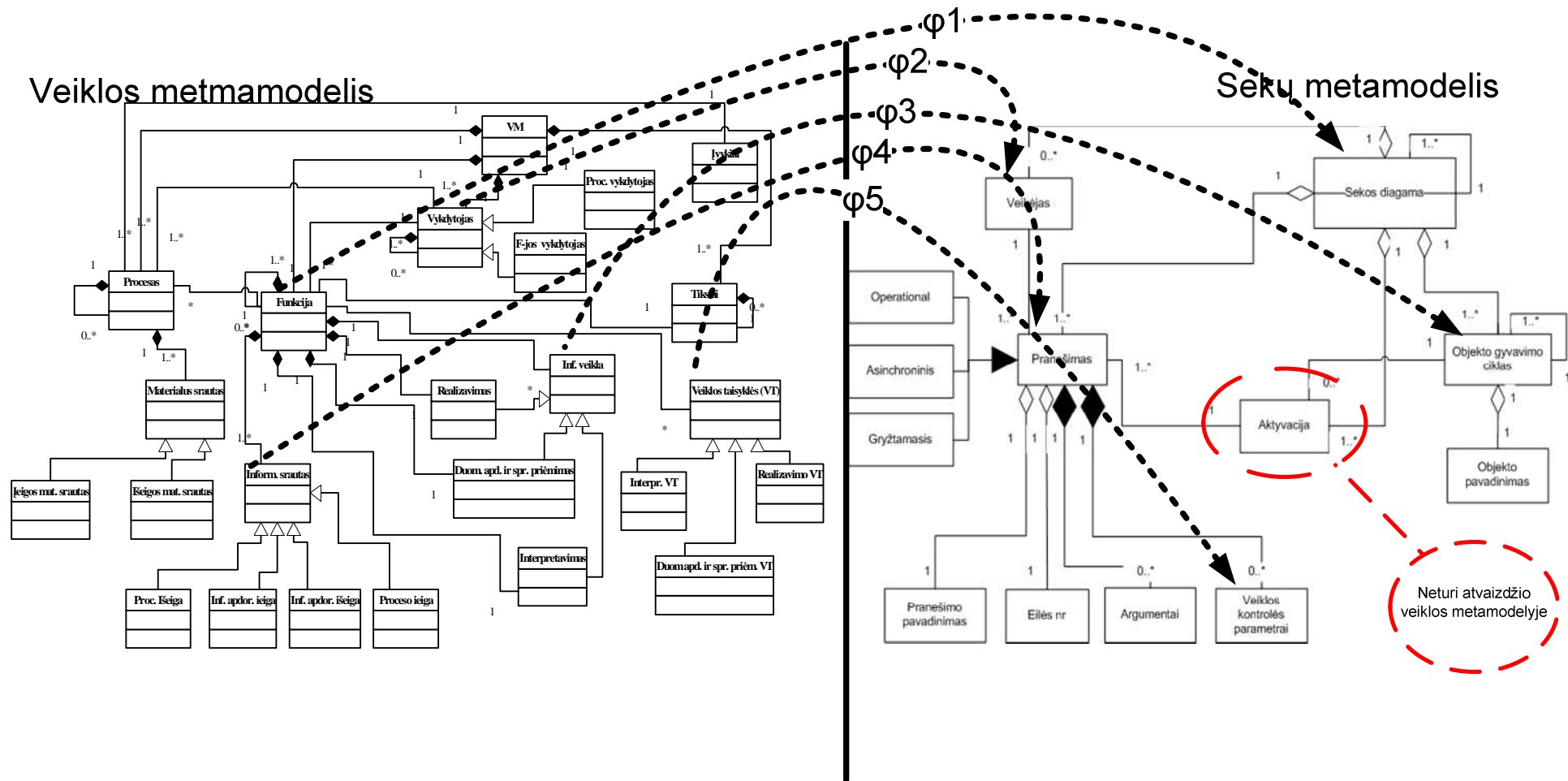
Veiklos metamodelio klasių atvaizdžiai į sekų metamodelį

Veiklos metamodelio klasės pavadinimas	Atvaizdis (→)	Sąveikos metamodelio klasės pavadinimas
VM.funkcija	φ1	SM.sekos diagrama
VM.Vykdytojas	φ2	SM.vykdytojas
VM.informacinė veikla	φ 3	SM.objekto gyvavimo ciklas
VM informacinis srautas	φ 4	SM.pranešimas
VM.veiklos taisyklės	φ 5	SM.veiklos kontrolės parametrai
Neturi atvaizdžio		SM.Aktyvacija

Šaltinis: sukurta autoriaus

22 paveiksle matome 5lentelėje formaliai aprašytus atvaizdžius pateiktus grafiškai. Kiekviena įmonėje atliekama funkcija yra aprašoma sekos diagrama φ1. Funkciją atliekantis vykdytojas atvaizduojamas kaip sekos diagramos vykdytojas sekos metamodelyje dekomponuojamų sekų iniciatorius φ2. Su funkcija susijusios informacinės veiklos yra atvaizduojamos objekto gyvavimo cikluose sekos metamodelyje φ 3. Informaciniai srautai kurių pagalba atliekama funkcija atitinka pranešimus sekos metamodelyje φ 4. Funkciją kontroliuojančios veiklos taisyklės veiklos metamodelyje atitinka veiklos kontrolės parametrus sekų metamodelyje φ 5. Sekų metamodelio elementas SM.Aktyvacija neturi atvaizdžio veiklos metamodelyje, todėl yra tikslinga atlikti veiklos metamodelio papildymą aprašytą 2.8. skryriuje.

22 paveiksle yra grafiškai pateikiamas sąveikos metamodelio atvaizdavimo proceso etapas, kurio metu susijusios veiklos metamodelio klasės yra atvaizduojamos sąveikos modelyje.

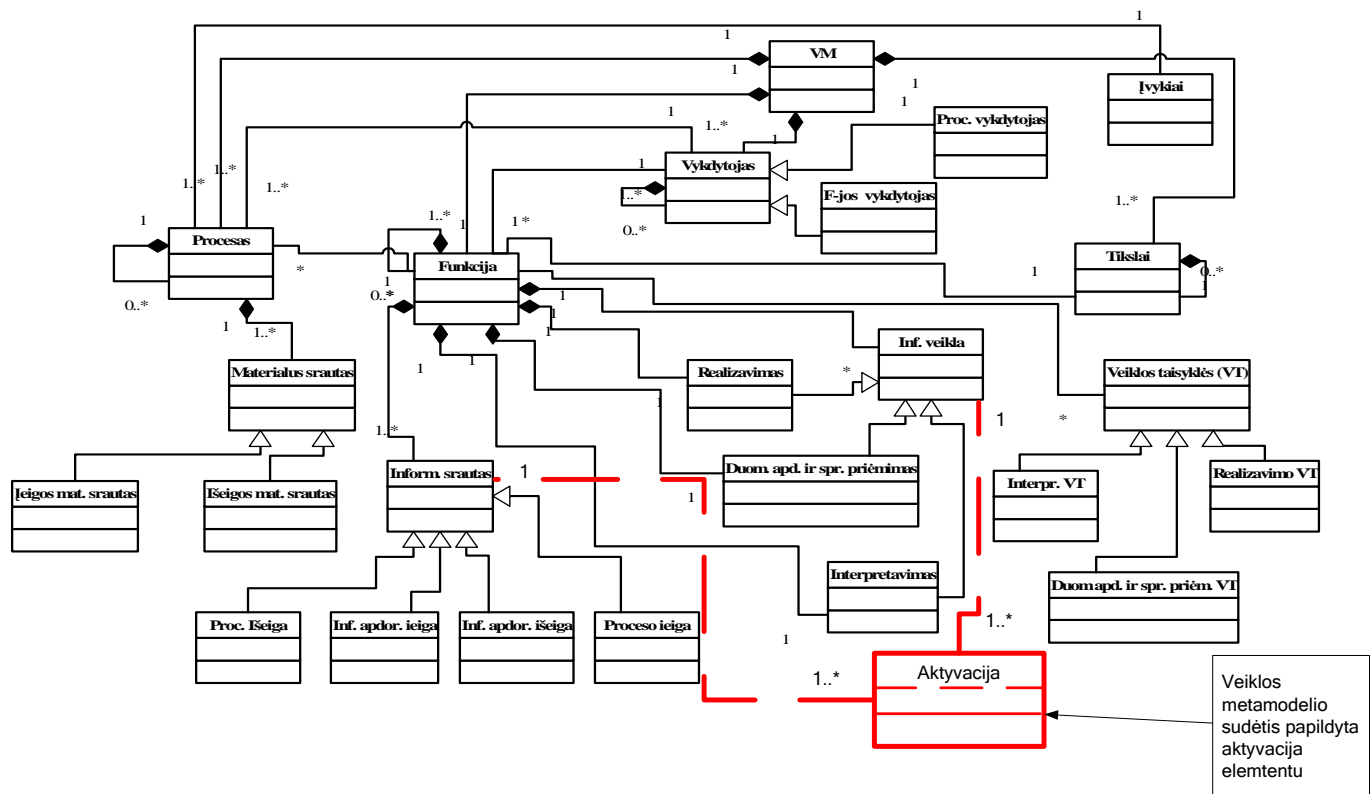


Šaltinis: sukurta autoriaus

22 pav. Veiklos metamodelio atvaizdavimas į UML sąveikų metamodelį

2.8 Veiklos metamodelio papildymas

Kaip matome 22 pav. Pateiktame veiklos metamodelio atvaizdavime į sekų metamodelį ne visi sekų metamodelio elementai turi atitikmenį esamame veiklos metamodelyje. Naudojant esamą veiklos metamodelį be atvaizdžio lieka sekų metamodelio elementas „Aktyvacija“ formaliai aprašomas kaip SM.Aktyvacija –elementas naudojamas nurodyti veiklos trukmę objekto gyvavimo cikle. Šis sekų modelio konstruktas yra svarbi modelio dedamoji dalis naudojama realaus laiko informacinės veiklos dekomponavime, veiklos funkcijų optimizavimo uždaviniuose (pavyzdžiui projektuojant serverio/kliento pagrindu paremtą IS reikia žinoti duomenų aktualumo bei atnaujinimo trukmę. Šią informaciją saugant veiklos modelyje ir panaudojant kituose modeliuose galima pritaikyti atitinkamą techninę įrangą, tuo optimizuojant laiko sąnaudas skirtas uždavinių įgyvendinimui). Todėl yra tikslinga į turimą veiklos metamodelį įtraukti trukmės klasę kaip informacinės veiklos sudedamąją dalį papildyta veiklos metamodelio klasių diagrama yra pateikiama 12pav.



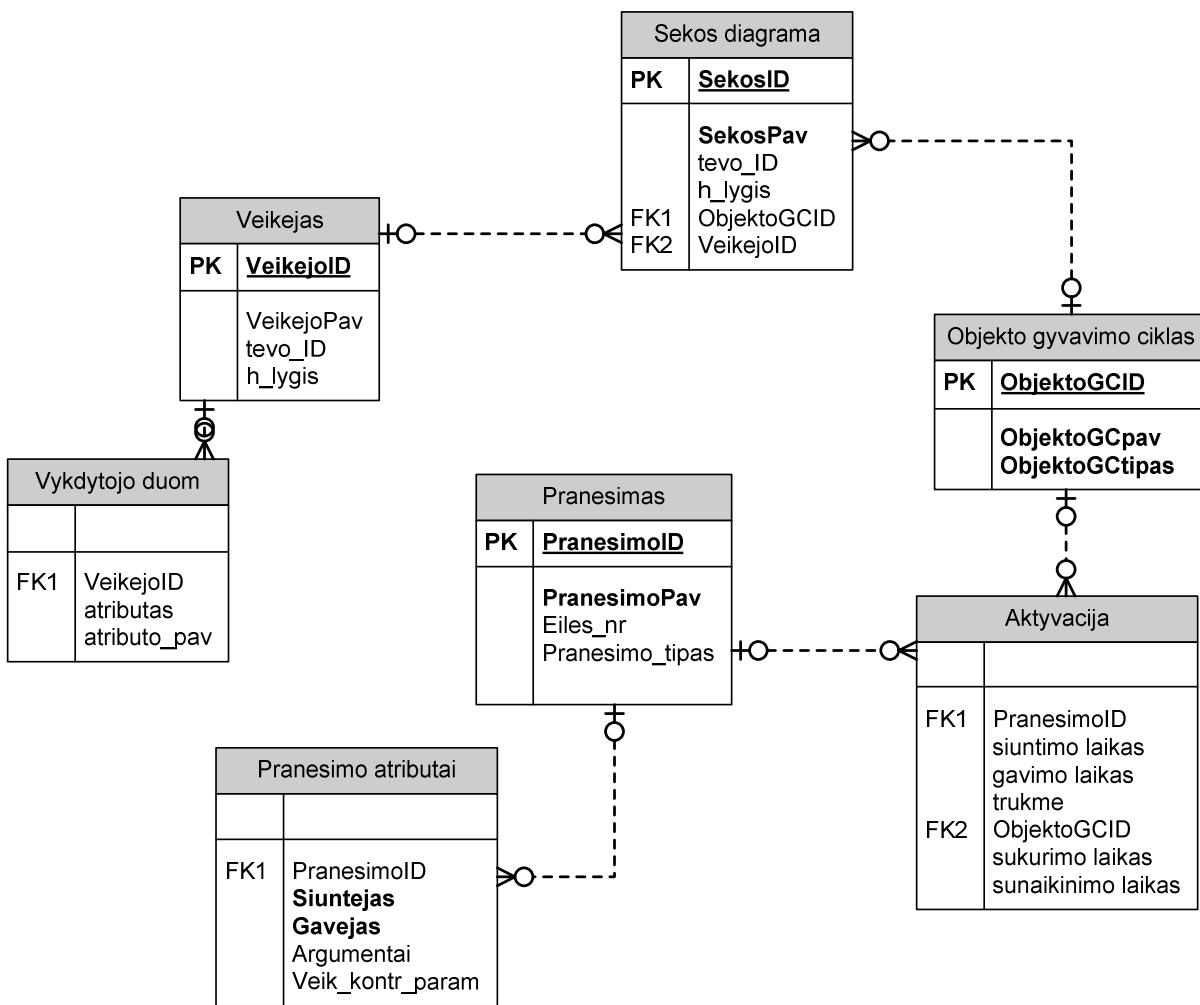
Šaltinis: sukurta autoriaus pagal Gudas S. Lopata A., 2005, Veiklos modelių grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikavimo metodas

23 pav. Papildytas veiklos metamodelis

3. VEIKLOS MODELIU GRINDŽIAMO UML SEKŲ MODELIO GENERAVIMAS

3.1 Siūloma sekų diagramos loginė duomenų bazės schema

Sekančiame paveiksle pateikiama sekų diagramos loginė duomenų bazės schema kuri aprašyta sekančiame skyriuje

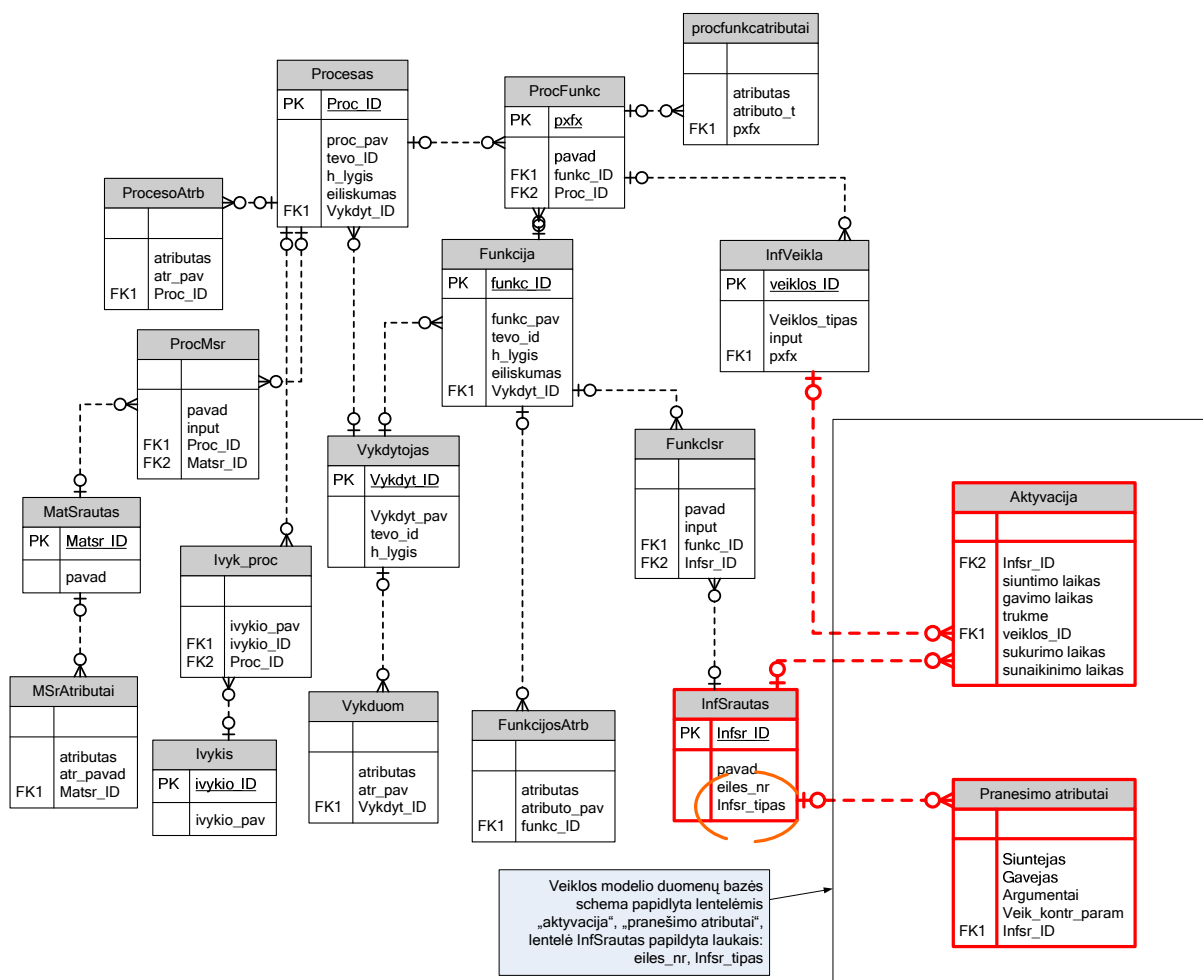


Šaltinis: sukurta autoriaus

24 Pav. Siūloma Sekų diagramos loginė duomenų bazės schema

3.2 Veiklos modelio loginės duomenų bazės schemos papildymas

Išanalizavus veiklos modelio ir sekų modelio loginės duomenų bazių schemas siekiant užtikrinti pilną sekų modelio atvaizdavimą buvo nuspręsta veiklos modelio duomenų bazės loginę schemą papildyti dviem lentelėmis: Aktyvacija ir InfSratributai, taip pat lentelę InfSrautas papildyti sekančiais laukais: eiles_nr, ir InfSr_tipas, kaip parodyta žemiau einančiame paveiksle.



Šaltinis: sukurta autoriaus pagal Gudas S. Lopata A., 2005, Veiklos modelių grindžiamas kompiuterizuotas funkcinio vartotojo reikalavimų specifikavimo metodas

25 pav. Veiklos modelio loginės duomenų bazės schemos papildymas

Toliau yra pateikiamas detalus veiklos modelio loginės duomenų bazės lentelių su atributų reikalingų sekų modelio generavimo papildymais detalus aprašymas.

6lentelė

Veiklos modelio db lentelės Procesas detalus aprašymas

Procesas		
Atributas	Tipas	Atributų aprašymas
Proc_id	Integer	Proceso unikalus numeris (identifikatorius)
Proc_pav	Tekstas	Proceso pavadinimas
Tevo_id	Integer	Proceso aukštesnio proceso („tėvo“) ID

Vykdyt_id	Integer	Vykdytojo, kuris vykdo procesą, ID
H_lygis	Integer	Proceso hierarchijos lygis
Eiliškumas	Integer	Proceso vykdymo numeris

Šaltinis: sukurta autoriaus

Lentelėje **Procesas** saugomi organizacijos procesai.

7lentelė

Veiklos modelio db lentelės ProcAtr detalus aprašymas

ProcAtr		
Atributas	Tipas	Atributų aprašymas
Proc_id	Integer	Proceso unikalus numeris (identifikatorius)
Atributas	Tekstas	Proceso atributo pavadinimas
Atr_pavad	Tekstas	Proceso atributo reikšmė

Šaltinis: sukurta autoriaus

Lentelėje **ProcAtr** saugomi proceso atributai.

8lentelė

Veiklos modelio db lentelės Funkcija detalus aprašymas

Funkcija		
Atributas	Tipas	Atributų aprašymas
Func_id	Integer	Funkcijos unikalus numeris (identifikatorius)
Func_pav	Tekstas	Funkcijos pavadinimas
Tevo_id	Integer	Funkcijos aukštesnės funkcijos („tėvo“) ID
Vykdyt_id	Integer	Vykdytojo, kuris vykdo funkcija, ID
H_lygis	Integer	Funkcijos hierarchijos lygis
Eiliškumas	Integer	Funkcijos vykdymo numeris

Šaltinis: sukurta autoriaus

Lentelėje **Funkcija** saugomos organizacijos funkcijos.

9lentelė

Veiklos modelio db lentelės FunkcijosAtrb detalus aprašymas

FunkcijosAtrb		
Atributas	Tipas	Atributų aprašymas
Func_id	Integer	Funkcijos unikalus numeris (identifikatorius)
Atributas	Tekstas	Funkcijos atributo pavadinimas
Atr_pavad	Tekstas	Funkcijos atributo reikšmė

Šaltinis: sukurta autoriaus

Lentelėje **FunkcijosAtrb** yra saugomi funkcijose atributai.

Veiklos modelio db lentelės Vykdytojas detalus aprašymas

Vykdytojas		
Atributas	Tipas	Atributų aprašymas
Vykdyt_id	Integer	Vykdytojo unikalus numeris (identifikatorius)
Vykdyt_pav	Tekstas	Vykdytojo pavadinimas
Tevo_id	Integer	Aukštesnio lygio vykdytojo („tėvo“) ID
H_lygis	Integer	Vykdytojo hierarchijos lygis

Šaltinis: sukurta autoriaus

Lentelėje **Vykdytojas** saugomi organizacijos skyriai, padaliniai, darbuotojai.

Veiklos modelio db lentelės VykDduom detalus aprašymas

VykDduom		
Atributas	Tipas	Atributų aprašymas
Vykdyt_id	Integer	Vykdytojo unikalus numeris (identifikatorius)
Atributas	Tekstas	Vykdytojo atributo pavadinimas
Atr_pavad	Tekstas	Vykdytojo atributo reikšmė

Šaltinis: sukurta autoriaus

Lentelėje **VykDduom** yra saugomi darbuotojų papildoma informacija. (telefonai, adresai)

Veiklos modelio db lentelės MatSrautas detalus aprašymas

MatSrautas		
Atributas	Tipas	Atributų aprašymas
Matsr_id	Integer	Materialaus srauto unikalus numeris (identifikatorius)
Pavad	Tekstas	Materialaus srauto pavadinimas

Šaltinis: sukurta autoriaus

Lentelėje **MatSrautas** yra saugomas materialus srautas (žaliavos, pusgaminiai ir t.t).

Veiklos modelio db lentelės MatsrAtrb detalus aprašymas

MatsrAtrb		
Atributas	Tipas	Atributų aprašymas
Matsr_id	Integer	Materialaus srauto unikalus numeris (identifikatorius)
Atributas	Tekstas	Materialaus srauto atributo pavadinimas
Atr_pavad	Tekstas	Materialaus srauto atributo reikšmė

Šaltinis: sukurta autoriaus

Lentelėje **MatsrAtrb** yra saugomi papildomi materialaus srauto atributai (pvz: ar žaliavos matuojamas kilogramais, ar metrais ir t.t).

Veiklos modelio db lentelės InfSrautas detalus aprašymas

InfSrautas		
Atributas	Tipas	Atributų aprašymas
InfSr_id	Integer	Informacinio srauto unikalus numeris (identifikatorius)
Pavad	Tekstas	Informacinio srauto pavadinimas
Eiles_nr	Tekstas	Informacinio srauto eilės numeris
InfSr_tipas	Tekstas	Informacinio srauto tipas

Šaltinis: sukurta autoriaus

Lentelėje **InfSrautas** yra saugomas informacinis srautas (pvz: dokumentacija einanti su materialiu srautu ir t.t).

Veiklos modelio db lentelės ProcMSr detalus aprašymas

ProcMSr		
Atributas	Tipas	Atributų aprašymas
Matsr_id	Integer	Materialaus srauto ID
Proc_id	Integer	Proceso ID
Pavad	Tekstas	Srauto pavadinimas
Input	Tekstas	Požymis ar materialus srautas yra proceso įėjimas ar išėjimas

Šaltinis: sukurta autoriaus

Lentelėje **ProcMSr** yra saugojamas procesų-materialaus srauto sąrašas. Jis parodo proceso naudojamą materialų srautą ir šio srauto tipą. (Ar tai proceso įėjimas ar išėjimas).

Veiklos modelio db lentelės FunkcIsr detalus aprašymas

FunkcIsr		
Atributas	Tipas	Atributų aprašymas
InfSr_id	Integer	Informacinio srauto ID
Funkc_id	Integer	Funkcijos ID
Pavad	Tekstas	Srauto pavadinimas
Input	Tekstas	Požymis ar informacinis srautas yra funkcijos įėjimas ar išėjimas

Šaltinis: sukurta autoriaus

Lentelėje **FunkcIsr** yra saugojamas funkcijų – informacinio srauto sąrašas. Jis parodo funkcijų naudojamą informacinį srautą ir šio srauto tipą. (Ar tai funkcijos įėjimas ar išėjimas).

Veiklos modelio db lentelės Ivykis detalus aprašymas

Ivykis		
Atributas	Tipas	Atributų aprašymas
Ivykio_id	Integer	Įvykio unikalus numeris (identifikatorius)
Ivykio_pav	Integer	Įvykio pavadinimas
Proc_id	Integer	Proceso ID

Šaltinis: sukurta autoriaus

Lentelėje **Ivykis** yra saugomi įvykiai: (Įvykis paleidžia konkrečius procesus).

Veiklos modelio db lentelės Ivyk_Proc detalus aprašymas

Ivyk_Proc		
Atributas	Tipas	Atributų aprašymas
Ivykio_id	Integer	Įvykio ID
Proc_id	Integer	Proceso ID
Pavad	Tekstas	Pavadinimas

Šaltinis: sukurta autoriaus

Lentelėje **Ivyk_Proc** yra saugomas įvykių – procesų sąrašas. Jis parodo kokie įvykiai sužadina procesus.

Veiklos modelio db lentelės ProcFunkc detalus aprašymas

ProcFunkc		
Atributas	Tipas	Atributų aprašymas
PxFx	Integer	Proceso-Funkcijos sankirtos unikalus numeris (identifikatorius)
Proc_id	Integer	Proceso ID
Funkc_id	Integer	Funkcijos ID
Pavad	Tekstas	Sankirtos pavadinimas

Šaltinis: sukurta autoriaus

Lentelėje **ProcFunkc** yra susiejami procesai su funkcijomis.

Veiklos modelio db lentelės ProcFunkcAtributai detalus aprašymas

ProcFunkcAtributai		
Atributas	Tipas	Atributų aprašymas
PxFx	Integer	Proceso-Funkcijos sankirtos ID (identifikatorius)
Atributas	Tekstas	Sankirtos atributas
Atributo_t	Tekstas	Sankirtos atributo tipas

Šaltinis: sukurta autoriaus

Lentelėje **PxFx** saugomi sankirtos atributai, bei jų tipai.

Veiklos modelio db lentelės Infveikla detalus aprašymas

Infveikla		
Atributas	Tipas	Atributų aprašymas
PxFx	Integer	Proceso-Funkcijos sankirtos ID (identifikatorius)
Veiklos_id	Integer	Informacinės veiklos (identifikatorius)
Input	Tekstas	Informacinės veiklos argumentai
Veiklos_tipas	Tekstas	Informacinės veiklos tipas

Šaltinis: sukurta autoriaus

Lentelėje Infveikla saugoma informacija apie proceso ir funkcijos sankirtos informacinę veiklą

Veiklos modelio db lentelės Aktyvacija detalus aprašymas

Aktyvacija		
Atributas	Tipas	Atributų aprašymas
Infsr_ID	Integer	Informacinio srauto unikalus numeris
Siuntimo_laikas	Date	Informacinio srauto siuntimo laikas
Gavimo_laikas	Date	Informacinio srauto gavimo laikas
Veiklos_ID	Integer	Informacinės veiklos unikalus numeris
Sukurimo_laikas	Date	Su informacine veikla sukurto objekto gyvavimo ciklo sukūrimo laikas
Sunaikinimo_laikas	Date	Su informacine veikla sukurto objekto gyvavimo ciklo sukūrimo laikas

Šaltinis: sukurta autoriaus

Lentelėje Aktyvacija saugoma informacija apie informacinės veiklos ir informacinio srauto sankirtą.

Veiklos modelio db lentelės Informacinio srauto atributai detalus aprašymas

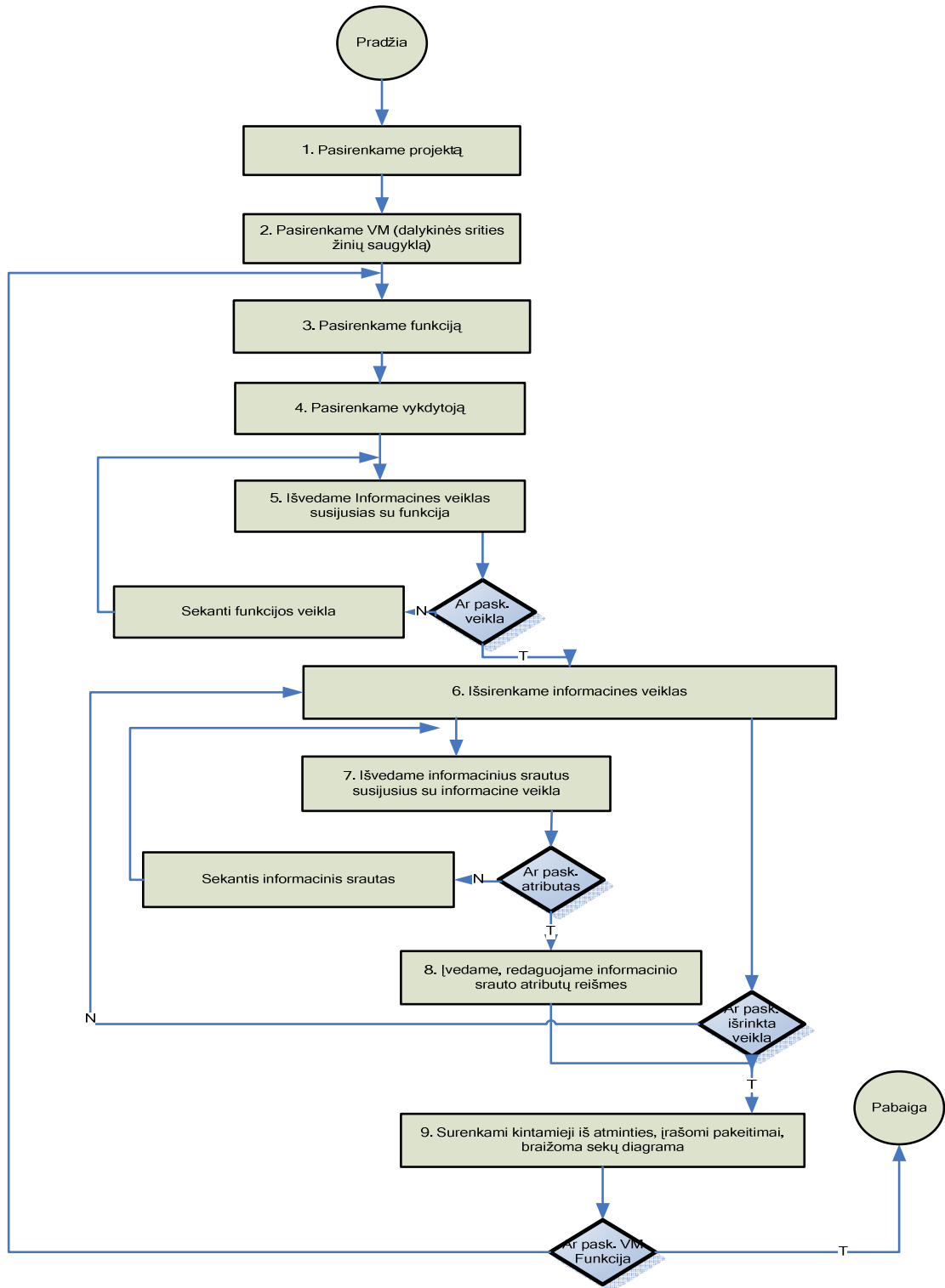
Informacinio srauto atributai		
Atributas	Tipas	Atributų aprašymas
Infsr_ID	Integer	Informacinio srauto unikalus numeris
Siuntejas	Tekstas	Informacinio srauto siuntėjas
Gavejas	Tekstas	Informacinio srauto gavėjas
Argumentai	Tekstas	Informacinio srauto Argumentai
Veiklos kontroles parametrai	Tekstas	Informacinio srauto kontrolės parametrai

Šaltinis: sukurta autoriaus

Lentelėje Informacinio srauto atributai saugoma informacija apie Informacinio srauto atributus

3.3 Siūlomas UML Sekų diagramos generavimo iš Veiklos modelio algoritmas

Sekančiame paveiksle pateikiamas UML sekų diagramos generavimo iš Veiklos modelio generavimo bazinis algoritmas



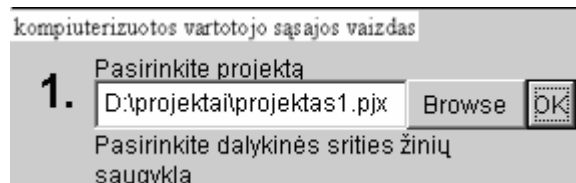
Šaltinis: sukurta autoriaus

26 pav. Siūlomas UML Sekų diagramos generavimo iš Veiklos modelio algoritmas

3.4 Siūlomas UML Sekų modelio generavimo iš veiklos modelio programos prototipas

Šiame skyriuje aprašomas programinės įrangos skirtos UML sekų modelio generavimui iš veiklos modelio prototipas paremtas 4.2 skyriuje pavaizduotu algoritmo žingsnius

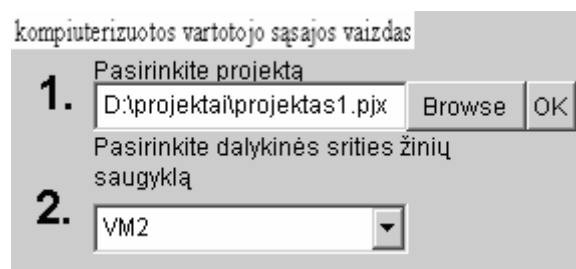
1 žingsnis



1. Grafinio redaktoriaus vaizdas
D:\projektai\projektas1.pjx

Kompiuterizuotoje vartotojo sąsajoje pasirenkame projektą, kuriame yra dalykinės srities saugyklos. Pasirinkus projektą sekų diagramoje yra užrašomas jo adresas kaip pavaizduota aukščiau esančiame grafinio redaktoriaus vaizde.

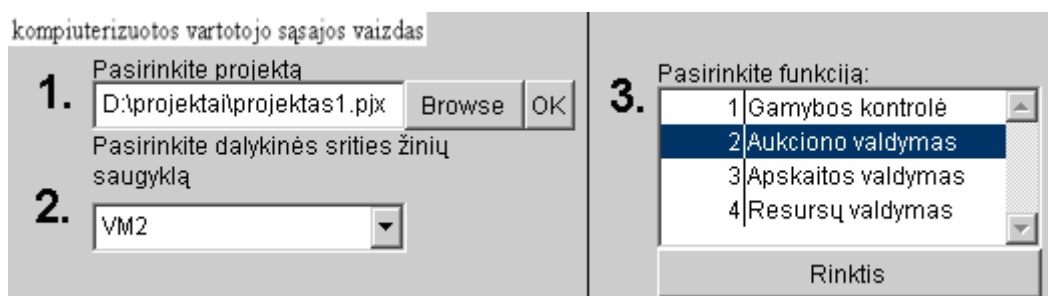
2 žingsnis



1. Grafinio redaktoriaus vaizdas
D:\projektai\projektas1.pjx
2. Dalykinės srities žinių saugykla: VM2

Pasirinkus projektą reikia pasirinkti dalykinės srities žinių saugyklą (veiklos modelį), kurioje yra saugomos žinios reikalingos sekų modeliui generuoti. Parinktas veiklos modelio pavadinimas yra atvaizduojamas sekų diagramoje kaip pavaizduota aukščiau esančiame grafinio redaktoriaus vaizde.

3 žingsnis



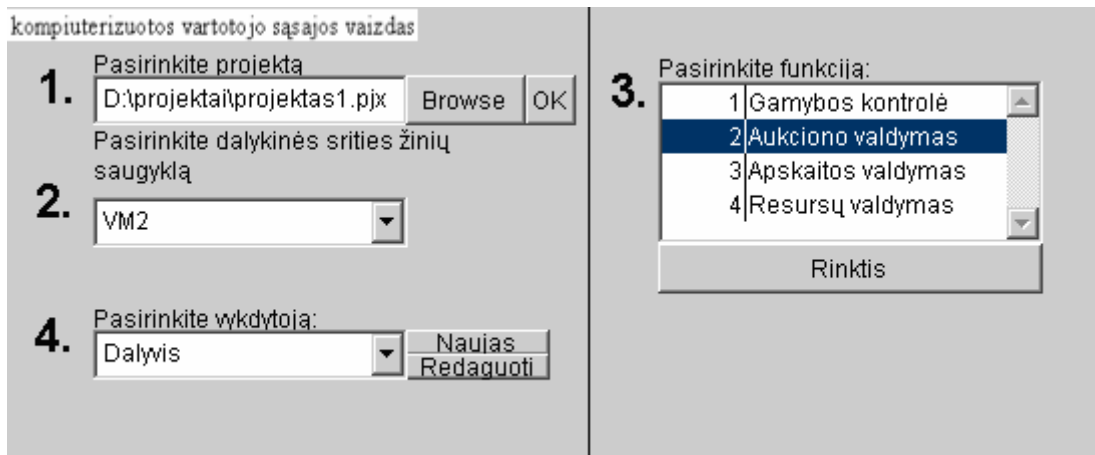
1. D:\projekta\projektas1.pjx
2. Dalykinės srities žinių saugykla:VM2

Grafinio redaktoriaus vaizdas

3. Funkcija: aukciono valdymas

Pasirinkus veiklos modelį lauką pažymėtą 3iuoju numeriu yra išvedamos visos funkcijos saugomos veiklos modelio duomenų bazės funkcija lentelėje, išvedami yra funkc_id ir funkc_pav laukai. Pasirinkus funkciją kuriai bus generuojama sekų diagrama jos pavadinimas yra atvaizduojamas sekų diagramoje kaip parodyta aukščiau esančiame grafinio redaktoriaus vaizde.

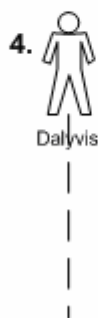
4žingsnis



1. D:\projekta\projektas1.pjx
2. Dalykinės srities žinių saugykla:VM2

Grafinio redaktoriaus vaizdas

3. Funkcija: aukciono valdymas



Pasirinkus funkciją parenkamas jos vykdytojas kaip parodyta aukščiau esančiame kompiuterizuotos vartotojo sąsajos vaizde 4-uoju numeriu pažymėtame lauke. Į ekraną yra išvedamas veiklos modelio duomenų bazės vykdytojas lentelės vykdyt_pav laukas. Pasirinkus funkcijos vykdytoją jis yra atvaizduojamas sekų diagramoje kaip parodyta aukščiau esančiame grafinio redaktoriaus vaizde.

5 žingsnis

kompiuterizuotos vartotojo sąsajos vaizdas

1. Pasirinkite projektą
D:\projektai\projektas1.pjx Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

3. Pasirinkite funkcija:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

Pasirinkus funkciją ir jos vykdytoją spaudžiamas 5-uoju numeriu pažymėtas mygtukas kompiuterizuotos vartotojo sąsajos lange. Paspaudus minėtą mygtuką pagal parinktą funkciją iš veiklos modelio duomenų bazėje saugomos informacijos yra išrenkamos informacinės veiklos susijusios su pasirinkta funkcija t.y lentelės InfVeikla laukai veiklos_id ir input. Jų atvaizdavimas matomas žemiau esančioje kompiuterizuotos vartotojo sąsajos formos lange.

6 žingsnis

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

kompiuterizuotos vartotojo sąsajos vaizdas

1. Pasirinkite projektą
D:\projektai\projektas1.pjx Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

3. Pasirinkite funkcija:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

6. Pasirinkite Objektus
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 Dalyvis
6 Objekto pristatymas
7 Dokumentacija

Rinktis

Aukščiau esančiame kompiuterizuotos vartotojo sąsajos lange 6-uoju numeriu pažymėtame lauke matome funkcijos informacinių veiklų atvaizdavimą. Iš išrinkto sąrašo išrenkame reikalingas veiklas sekų diagramos generavimui.

7 žingsnis

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

kompiuterizuotos vartotojo sąsajos vaizdas

1. Pasirinkite projektą
D:\projekta\projektas1.pjx
Pasirinkite dalykinės srities žinių saugyklą

2. VM2

4. Pasirinkite vykdytoją:
Dalyvis

3. Pasirinkite funkciją:
1 Gamybos kontrolė
2 **Aukciono valdymas**
3 Apskaitos valdymas
4 Resursų valdymas

5.

6. Pasirinkite Objektus

1	Aukciono sąsaja
2	Dalyvių sąrašas
3	Objekto aprašai
4	Siūlymų laikmatis
5	Dalyvis

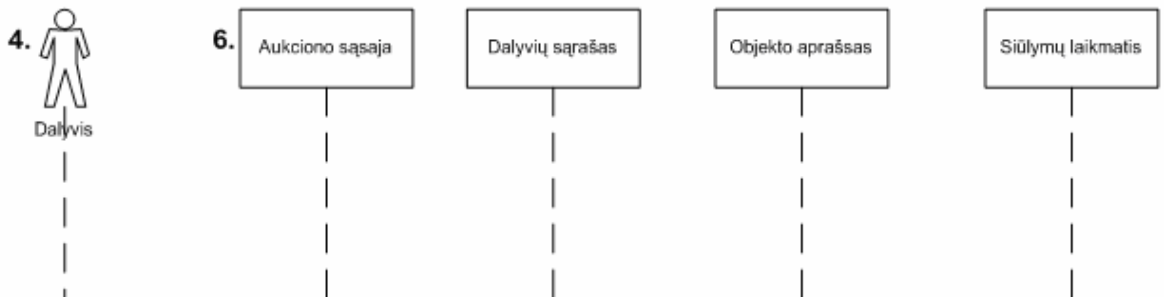
7.

1. D:\projekta\projektas1.pjx

Grafinio redaktoriaus vaizdas

2. Dalykinės srities žinių saugykla: VM2

3. Funkcija: aukciono valdymas



Pasirinkę informacines veiklas kurios bus atvaizduojamos kaip objektai sekų diagramoje spaudžiame mygtuką „rinktis“ pažymėtą 7-uoju numeriu aukščiau esančiame kompiuterizuotos vartotojo sąsajos lange. Šiuo veiksmu iš išvesto sąrašo pašaliname nereikalingas veiklas ir atvaizduojame jas kaip objekto gyvavimo ciklus sekų diagramoje kaip parodyta aukščiau esančiame paveiksle.

8.1. žingsnis

kompiuterizuotos vartotojo sąsajos vaizdas

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

1. Pasirinkite projektą
D:\projektai\projektas1.pjx Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

3. Pasirinkite funkciją:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

6. Pasirinkite Objektus:
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 Dalyvis

7. Rinktis

8.1 Informaciniai srautai:

Pranesimo_id	Eiles_nr	Pavadinimas	Tipas	Siuntimo_laikas	Gavejas	Gavimo_laikas	Trukme	Argumentai	Parametrai
1	1	Sukurti naują aprašą	oper	05.16.2007 12:00	Objekto aprašas	05.16.2007 12:01	1min		create new()
4	4	vesti naują siūlymą	oper	05.16.2007 13:00	dalyvių sąrašas	05.16.2007 13:05	5min		newbidder()
13	12	Vieta pasibaigusiame aukcione	gryzt	05.18.2007 13:15	dalyvis	05.18.2007 13:20	5min		

Du kartus spragtelėję ant aukščiau esančiame paveiksle 6-uoju numeriu pažymėtame lauke esančio įrašo į 8-uoju numeriu pažymėtus laukus išrenkame su pasirinktu objektu susijusius informacinius srautus. Iš veiklos modelio duomenų bazės yra išrenkami ir išvedami sekantys laukai: iš lentelės „informacinis srautas“ laukai „Infsr_id“, „pavad“, „eiles_nr“, „infsr_tipas“; iš lentelės „pranešimo atributai“ išvedami laukai „gavėjas“, „argumentai“, „veikl_kontr_param“ (laukui „siuntėjas“ automatiškai priskiriamas 6-jame lauke parinkta informacinė veikla); iš lentelės „Aktyvacija“ laukai: „siuntimo laikas“, „gavimo laikas“, „trukmė“ (laukams „sukūrimo laikas“ ir „gavimo laikas“ yra priskiriami anksčiausias ir vėliausias pranešimo siuntimo ir gavimo laukai atitinkamai). Aukščiau esančiame paveiksle 8.1 lauke matome išvestus objekto aukciono sąrašas informacinius srautus. Kurie gali būti kuriami, redaguojami, trinami tiesiog formoje esančiuose laukuose.

8.2 žingsnis

kompiuterizuotos vartotojo sąsajos vaizdas

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

1. Pasirinkite projektą
D:\projektai\projektas1.pix Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

3. Pasirinkite funkciją:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

6. Pasirinkite Objektus
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 Dalyvis

7. Rinktis

8.2 Informaciniai srautai:

Pranesimo_id	Eiles_nr	Pavadinimas	Tipas	Siuntimo_laikas	Gavejas	Gavimo_laikas	Trukme	Argumentai	Parametrai
	5	patikrinti pasiūlymą	asyn	07.16.2007 13:05	Dalyviu sarasas	07.16.2007 13:06	1min		valid bidder()
	6	padidinti esamą kainą	oper	07.16.2007 13:16	Objekto aprašas	07.16.2007 13:17	1min	kaina	if valid bidder(true),bidup()
	7	pasiūlymas netinkamas	gryzt	07.16.2007 13:16	dalyvis	07.16.2007 13:17	1min		if valid bidder(false)
	12	aukciono laimetojas	oper	07.18.2007 12:07	aukciono sąsaja	07.18.2007 12:08	1min	laimetojas	

Šiame žingsnyje į aukščiau esančiame paveiksle 8.2 numeriu pažymėtus laukus įvedame, redaguojame, papildome objekto „dalyvių sąrašas“ informacinius srautus išrinktus pagal 8.1 žingsnyje aprašytą eigą.

8.3. Žingsnis

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

kompiuterizuotos vartotojo sąsajos vaizdas

1. Pasirinkite projektą
D:\projektai\projektas1.pjx Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

3. Pasirinkite funkciją:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

6. Pasirinkite Objektus
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 Dalyvis

7. Rinktis

8.3 Informaciniai šaltiniai:

Pranesimo_id	Eiles_nr	Pavadinimas	Tipas	Siuntimo_laikas	Gavejas	Gavimo_laikas	Trukme	Argumentai	Parametrai
	2	Nustatyti laikmatį	oper	05.16.2007 12:01	siūlymų laikmatis	05.16.2007 12:02	1min	48h	settimer()
	8	Išsaugoti kainą	oper	05.16.2007 13:20	Objekto aprašas	05.18.2007 11:59	46h 39min		for each bidup(), savebid()
	9	Išsaugoti kainą	oper	05.16.2007 13:20	Objekto aprašas	05.18.2007 11:59	46h 39min		do until nomorebids()
11	10	nustatyti aukščiausią kainą	oper	05.18.2007 12:01	dalyvių sąrašas	05.18.2007 12:05	3min		getwinningbit()

Šiame žingsnyje į aukščiau esančiame paveiksle 8.3 numeriu pažymėtus laukus įvedame, redaguojame, papildome objekto „objekto aprašas“ informacinius srautus išrinktus pagal 8.1 žingsnyje aprašytą eigą.

8.4. Žingsnis

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

kompiuterizuotos vartotojo sąsajos vaizdas

1. Pasirinkite projektą
D:\projektai\projektas1.pjx Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

3. Pasirinkite funkciją:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

6. Pasirinkite Objektus
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 Dalyvis

7. Rinktis

8.4 Informaciniai šaltiniai:

Pranesimo_id	Eiles_nr	Pavadinimas	Tipas	Siuntimo_laikas	Gavejas	Gavimo_laikas	Trukme	Argumentai	Parametrai
	10	Aukciono pabaiga	oper	05.18.2007 12:00	Objekto aprašas	05.18.2007 12:01	1min		timeElapsed()

Šiame žingsnyje į aukščiau esančiame paveiksle 8.4 numeriu pažymėtus laukus įvedame,

redaguojame, papildome objekto „siūlymų laikmatis“ informacinius srautus išrinktus pagal 8.1 žingsnyje aprašytą eigą

8.5. Žingsnis

kompiuterizuotos vartotojo sąsajos vaizdas

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

1. Pasirinkite projektą
D:\projektai\projektas1.pjx Browse OK

Pasirinkite dalykinės srities žinių saugyklą

2. VM2

3. Pasirinkite funkcija:
1 Gamybos kontrolė
2 Aukciono valdymas
3 Apskaitos valdymas
4 Resursų valdymas

5. Rinktis

4. Pasirinkite vykdytoją:
Dalyvis Naujas Redaguoti

6. Pasirinkite Objektus
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 Dalyvis

7. Rinktis

8.5 Informaciniai srautai:

Praesimo_id	Eiles_nr	Pavadinimas	Tipas	Siuntimo_laikas	Gavejas	Gavimo_laikas	Trukme	Argumentai	Parametrai
3	3	siulyti kainą	oper	07.16.2007 12:15	aukciono sąsaja	07.16.2007 12:16	1	150	

Šiame žingsnyje į aukščiau esančiame paveiksle 8.5 numeriu pažymėtus laukus įvedame, redaguojame, papildome su funkcijos vykdytoju „dalyvis“ susijusius informacinius srautus išrinktus pagal 8.1 žingsnyje aprašytą eigą

9. Žingsnis

UML SEKŲ DIAGRAMOS GENERAVIMO MODULIS

kompiuterizuotos vartotojo sąsajos vaizdas

1. Pasirinkite projektą
D:\projektai\projektas1.pjx

2. Pasirinkite dalykinės srities žinių saugyklą
VM2

3. Pasirinkite funkciją:
1 Gamybos kontrolė
2 **Aukciono valdymas**
3 Apskaitos valdymas
4 Resursų valdymas

4. Pasirinkite vykdytoją:
Dalyvis

5.

6. Pasirinkite Objektus
1 Aukciono sąsaja
2 Dalyvių sąrašas
3 Objekto aprašai
4 Siūlymų laikmatis
5 **Dalyvis**

7.

8.5. Informaciniai šaltiniai:

Prasimo_id	Eiles_nr	Pavadinimas	Tipas	Siuntimo_laikas	Gavejas	Gavimo_laikas	Trukme	Argumentai	Parametrai
	3	3 siulyti kainą	oper	07.16.2007 12:15	aukciono sąsaja	07.16.2007 12:16	1	150	

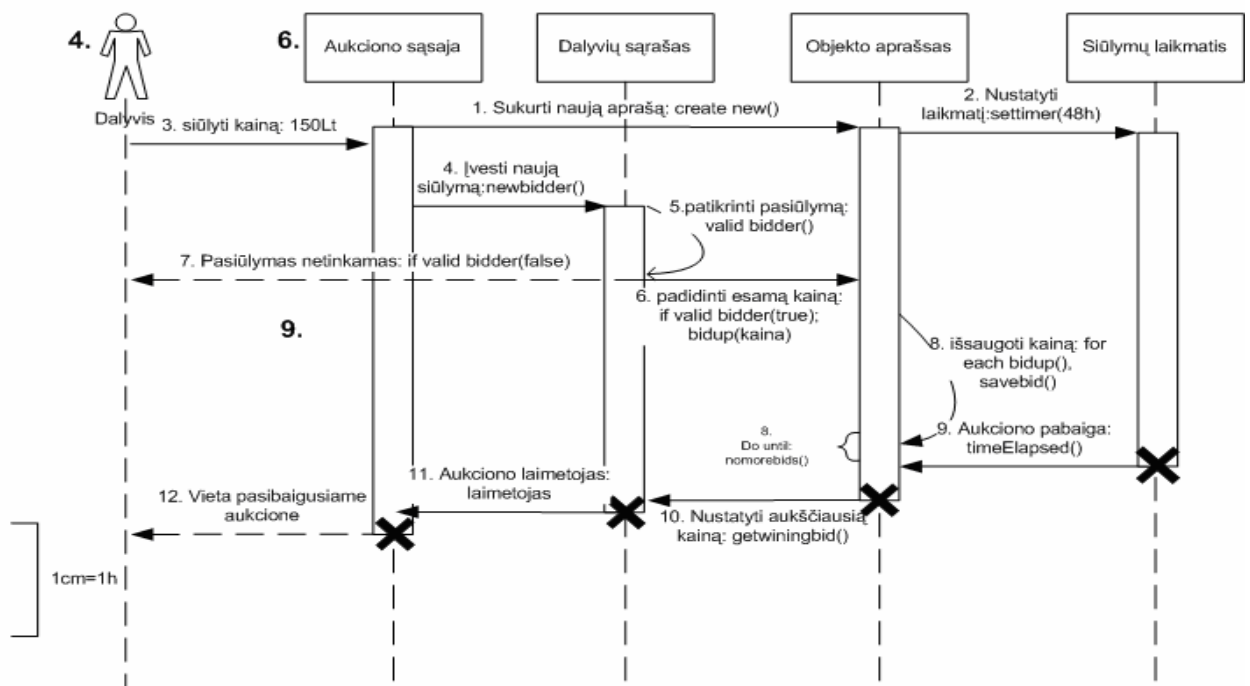
9.

1. D:\projektai\projektas1.pjx

Grafinio redaktoriaus vaizdas

2. Dalykinės srities žinių saugykla: VM2

3 Funkcija: aukciono valdymas



Suvedus išrinkus paredagavus visą reikiamą informaciją apie informacinius srautus susijusius su informacinėmis veiklomis spaudžiamas aukščiau esančiame paveiksle 9uoju numeriu pažymėtas mygtukas „generuoti sekų diagramą“. Paspaudus mygtuką iš laikinosios atminties(memvar) yra surenkami ir į duomenų bazės lenteles įrašomi nauji ir redaguoti laukai. Veiklos modelio duomenų bazės lentelės „informacinis srautas“ laukai „Infsr_id“, „pavad“, „eiles_nr“, „infsr_tipas“, iš lentelės „pranešimo atributai“ laukai: „gavėjas“, „argumentai“, „veikl_kontr_param“(laukui „siuntėjas“ automatiškai priskiriamas 6-jame lauke parinkta informacinė veikla); iš lentelės „Aktyvacija“ laukai: „siuntimo laikas“, „gavimo laikas“, „trukmė“ (laukams „sukurimo laikas“ ir „gavimo laikas“ yra priskiriami anksčiausias ir vėliausias pranešimo siuntimo ir gavimo laukai atitinkamai). Jeigu informacija yra validi generuojama sekų diagrama tarp ankstesniuose žingsniuose nubrėžtų vykdytojo ir objekto gyvavimo ciklą brėžiami informaciniai srautai (pranešimai), su savo argumentais ir parametrais, pagal tam tikrą užsiduotą lauko mastelį (pvz.: 1cm=1h) gavus pirmąjį pranešimą yra sukuriamas objekto aktyvacija išsiuntus paskutinį pranešimą aktyvacija yra ištrinama. Sugeneravus diagramą gaunama funkcijos aukciono valdymas sekų diagrama matoma aukščiau esančiame grafinio redaktoriaus vaizde. Jei tai paskutinė funkcija kuriai generuojama diagrama darbą baigiame, jei ne kartojame žingsnius nuo 3-čiojo.

IŠVADOS

1. Išanalizavus Žiniomis grindžiamos inžinerijos IV etapo pagrindus bei nustačius veiklos modelio vietą joje buvo nustatyta, kad norint sugeneruoti UML sekų diagramą reikia sukurti sekų diagramos generavimo algoritmą bei programinės įrangos paketą įgyvendinantį minėtą algoritmą.

1.1 Atlikus pagrindinių veiklos modeliavimo standartų, metodologijų bei modeliavimo kalbų analizę buvo nuspręsta magistro baigiamajame darbe naudoti „Gudas Lopata veiklos modelio“ metodą kaip pagrindą UML sekų diagramos generavimo proceso kūrimui. Kadangi kitos analizuotos veiklos modeliavimo metodologijos neatsižvelgia į sprendimo priėmimo požiūrį, silpnai išvystytas organizacinis požiūris Gudas Lopata veiklos modelio metodas turi pilnai apibrėžtus visus analizuotus požiūrius. Taip pat išsiskiria modeliavimo konstrukčių skaičiumi 19 kai tuo tarpu kitos metodologijos turi apie 10.

1.2 Išanalizavus populiariausių šiuo metu CASE įrankių analizė buvo nustatyta jog jų projektinių modelių generavimo galimybės yra nepakankamos, nes dauguma programinių paketų: Oracle Developer 10g, System Architect orientuojasi į kodo, dokumentacijos generavimą bei atgalinę inžineriją. Tik Provision Workbench 4.0 turi veiklos modelį kuriame saugomos dalykinės srities žinios. Visi kiti programiniai paketai įgyvendina tik diagramų generavimą vieną iš kitos arba dalinį jų generavimą. Programa Magic Draw UML7.1 iš vis nenumato projektinių modelių kūrimo automatizavimo galimybės visi modeliai kuriami nepriklausomai vienas nuo kito. Todėl buvo nuspręsta šiame darbe kurti naują UML sekų diagramos generavimo iš veiklos modelio procesą automatizuojantį algoritmą ir jį pagrindžiantį programinės įrangos prototipą.

2. Atlikus tyrimą buvo nustatyta, kad veiklos modelis yra svarbi žinių posistemio dalis, kuriame sukauptos dalykinės srities žinios gali būti naudojamos automatizuotam projektinių modelių generavimui.

2.1 Išanalizavus veiklos metamodelio ir sekų metamodelio sudėtį nustatyta, kad veiklos metamodelyje esančių žinių nepakanka sekų modelio generavimui, todėl jį tikslinga papildyti nauju elementu „aktyvacija“.

2.2 Siekiant užtikrinti pilną sekų modelio struktūros saugojimą į veiklos modelį, veiklos modelio loginė duomenų bazės schema papildyta dviem lentelėmis: „Aktyvacija“ ir „Informacinio srauto atributai“, taip pat papildyta lentelė „informacinis srautas“ naujais laukais: „eilės numeris“ ir „informacinio srauto tipas“.

- 3.** Sukurtas UML sekų modelio generavimo algoritmas, intelektualizuojantis IS inžinerijos projektavimo etapą tuo, kad automatizuojamas UML sekų diagramos kūrimo procesas.
- 4.** Siekiant patikrinti algoritmo funkcionalumą sukurtas sekų diagramos generavimą imituojantis programinės įrangos prototipas, kuris sėkmingai išbandytas taikant realius duomenis.
- 5.** Atlikus mokslinį tyrimą, nustatyta kad sekų modelio generavimas intelektualizuoja IS inžinerijos projektavimo etapą, nes sutaupomas projektuotojo laikas ir sumažinama klaidos tikimybė, kadangi į veiklos modelį surinktos žinios yra patikrintos valdymo teorijos požiūriu.

LITERATŪROS SĄRAŠAS

1. Gudas S., Skersys T., Lopata A. (2004) Framework for knowledge-based IS engineering, LNCS 3261, Proceedings of Third International Conference “Advances in Information Systems” (ADVIS 2004), Springer Berlin p. 512- 522. ISSN 0302-9743, ISBN 3-540-23478-0 (Gudas S., Skersys T., Lopata A. 2004)
2. Gudas S. Lopata A. (2005) Veiklos modelių grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikuojimo metodas, mokslo daktaro laipsnio disertacija, VU KHF. (Gudas S., Lopata A. 2005)
3. Gudas S., Lopata A. (2005) Žiniomis grindžiamos informacijos sistemų inžinerijos bruožai, VU KHF
4. Lopata A. (2002) Veiklos modelių sudėties analizė. Informacinės Technologijos konferencijos pranešimų medžiaga. Kaunas: Technologija p.377-381
5. Gudas S., Lopata A., Skersys T. (2002) Domain Knowledge Integration For Information Systems Engineering „Informacinės technologijos verslui 2002: [konferencijos pranešimų medžiaga]. Kaunas: Technologija 2002. p.56-59
6. Miller R (2003) Practical UML: A Hands-On Introduction for Developers [interaktyvus], [žiūrėta 2005 11 30] Prieiga per internetą: <http://bdn.borland.com/article/0,1410,31863,00.html>
7. Object Management Group, OMG (2003) Unified Modeling Language specification, version 1.5,[interaktyvus] [žiūrėta 2005 11 30]prieiga per Internetą:<http://www.omg.org/cgi-bin/doc?formal/03-03-01>
8. Larman C. (2005) UML interaction diagrams [interaktyvus], Indianapolis [žiūrėta 2005 10 30]. Prieiga per internetą: <http://www.informit.com/articles/article.asp?p=360441&seqNum=2> ,
9. Louisiana State University (2005) Interaction diagrams: business process reengineering project done in partial fulfillment of requirements for ISDS 7525 [interaktyvus], [žiūrėta 2005 11 01]. Prieiga per internetą <http://isds.bus.lsu.edu/cvoc/learn/bpr/cprojects/spring1998/modeling/interaction.html>,
10. Nuno Jardim Nunes, João Falcão e Cunha (2000) Towards a UML profile for interaction design: the Wisdom approach[interaktyvus] [žiūrėta 2005.10.30] prieiga per internetą: http://math.uma.pt/tupis00/submissions/nunes_Cunha/nunescunha.html,
11. Michaël Petit (2002) Report on the State of the Art in Enterprise Modelling [interaktyvus] University of Namur [žiūrėta 2007 03 10]. Prieiga per internetą: <http://athena.troux.com/akmii/Default.aspx?WebID=249>
12. CIMOSA (1996) A Primer on key concepts, purpose and business value [interaktyvus] [žiūrėta 2007 03 10]. Prieiga per internetą <http://cimosacnt.pl/Docs/Primer/primer5.htm>
13. Laurent Ferrier, Patrick Heymans and Michaël Petit (2003) Some Hints for a Clarification of CEN ENV 12204, [interaktyvus] Computer Science Department, University of Namur [žiūrėta 2007 03 10] prieiga per internetą: <http://www.info.fundp.ac.be/~mpe/papers/EEEI00abstract.doc>
14. IFAC-IFIP Task Force (2001) Universal Enterprise modeling language [interaktyvus] UEML group [žiūrėta 2007 03 15]. Prieiga per internetą: <http://www.cit.gu.edu.au/~bernus/taskforce/archive/UEML-TF-IG.ppt>
15. David Shorter (2004) Modeling and Architecture Work program and key resources [interaktyvus] ISO TC184 SC5 WG1 [žiūrėta 2007 03 15]. Prieiga per internetą: <http://www.mel.nist.gov/sc5wg1/wg1-on-a-page.pdf>
16. Maria Bergholtz, Paul Johannesson, Petia Wohed (2004) UEML: Providing Requirements and Extensions for Interoperability Challenges.[interaktyvus] Department of Computer and Systems Sciences Stockholm University [žiūrėta 2007 04 02]. Prieiga per internetą: <http://people.dsv.su.se/~petia/Publications/EI2N05.pdf>

17. K.Kosanke (1996) Comparison of Modelling Methodologies [interaktyvus] CIMOSA Association [žiūrėta 2007 04 02] prieiga per internetą: <http://cimoso.cnt.pl/Docs/cmm.htm>
18. Ovidiu S. Noran (2000) Business Modelling: UML vs. IDEF [interaktyvus] Griffith University School of Computing and Information [žiūrėta 2007 04 02]. Prieiga per Internetą: http://www.cit.gu.edu.au/~noran/cit_6114
19. Saonee Sarker (2000) Tutorial for Oracle designer [interaktyvus] [žiūrėta 2007 04 10]. Prieiga per internetą: <http://www.prenhall.com/divisions/bp/app/hoffer/oracle/tutorial/>
20. ProVision - Enterprise v6.0 (2007) Proforma corp. [demonstracinė produkto versija] [žiūrėta 2007 04 12]. Prieiga per internetą. <http://www.proformacorp.com/Downloads/evaluations.asp?login=false>
21. System Architect (2007) Telelogic [demonstracinė produkto versija] [žiūrėta 2007 04 12]. Prieiga per internetą: http://www.telelogic.com/company/events/webinars/ondemand_player/index.cfm?id=4457
22. Magic Draw UML 7.0 [demonstracinė produkto versija] [žiūrėta 2007 04 15]. Prieiga per internetą: http://www.magicdraw.com/main.php?ts=navig&NMSESSID=5d5d55b302c2918212ed2dcd071e2ef9&cmd_show=1&menu=online_demo&NMSESSID=5d5d55b302c2918212ed2dcd071e2ef9

PRIEDAI

1. Priedas „Informacinės Technologijos 2006“ Konferencijos pranešimas

VEIKLOS MODELIU GRINDŽIAMAS UML SEKŲ MODELIO GENERAVIMAS

magistrantas Š.Čirvinskas

El.p: sarunas.c@dokeda.lt

dr. A. Lopata

El.p Audrius.lopata@ktu.lt

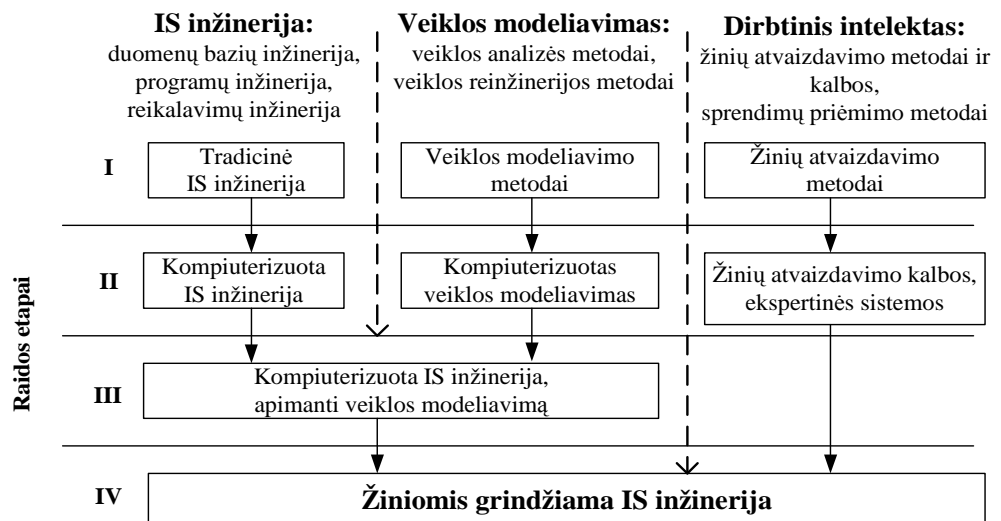
Vilniaus Universitetas

Kauno Humanitarinis Fakultetas

Straipsnyje yra apžvelgiama žiniomis grindžiamos IS inžinerijos raida, aprašomas veiklos metamodelis, pateikiama UML diagramų klasifikacija Pagrindinis atlikto tyrimo etapas - UML sąveikos modelio generavimo proceso analizė, pagal pateiktą veiklos metamodeliu

1. Žiniomis grindžiama IS inžinerija

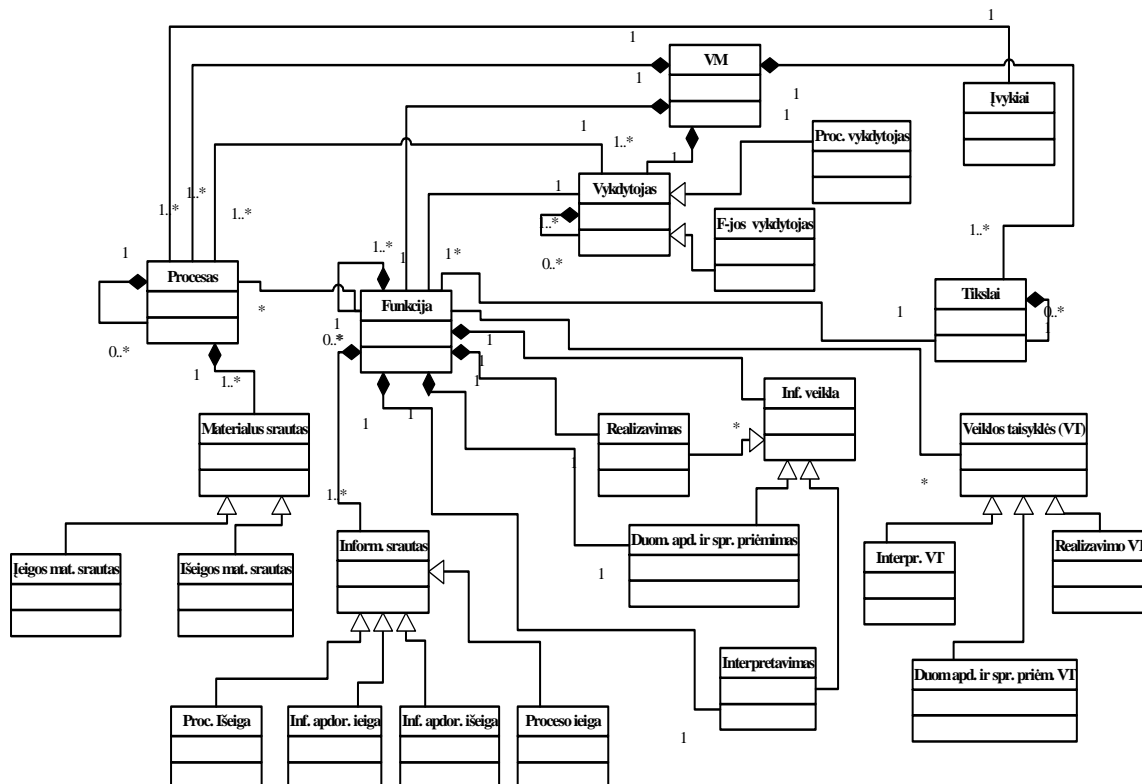
Pastaraisiais metais kompiuterizuota IS inžinerija (CASE) yra pasiekusi naują kokybinį vystymosi lygį – ji tampa žiniomis grindžiama IS inžinerija. Reikia pabrėžti, kad jau du dešimtmečiai informacinės sistemos yra kuriamos CASE sistemų (angl. *Computer Aided Systems Engineering tools*) aplinkose. Iš viso šiuo metu priskaičiuojama apie šimtą skirtingų CASE sistemų. Iš pradžių (žr. 1 pav.) funkcionavo tik tradicinė IS inžinerija, kurią vėliau keitė kompiuterizuota IS inžinerija. Dar vėliau į kompiuterizuotą IS inžineriją buvo integruotas veiklos modeliavimas. Galiausiai informacinių technologijų vystymasis, informacijos svarba šiuolaikiniame versle, naujų verslo modeliavimo priemonių kūrimas paskatino naują žiniomis grindžiamą CASE sistemą. Ketvirtajame IS inžinerijos etape CASE sistemos dalimi tapo žinių posistemis, kuriame yra kaupiamos būtinos žinios apie veiklos sritį (organizacijos procesus ir funkcijas). Žinių posistemio struktūrą apibrėžia formalizuota žinių apie veiklos valdymą struktūra - veiklos metamodelis (3). Šiame darbe bus analizuojamas vienas iš ketvirtojo IS inžinerijos vystymosi etapo aspektų: UML sekų (angl. Sequence) modelio generavimas remiantis turimu veiklos metamodeliu.



1 pav. Pagrindiniai IS inžinerijos raidos etapai (3)

2. Veiklos metamodelis

Egzistuoja įvairūs metamodeliai specifikuojantys dalykinės srities žinių saugojimo struktūrą: CEN ENV 12204(10), CEN ENV 40003(11), GERAM(6).(1) Veiklos metamodelyje saugomos žinios apie dalykinę sritį kurios būtinos projektinių modelių ir programinio kodo generavimui. 2 paveiksle pateiktas veiklos metamodelio, sukurtas Kauno Technologijos universiteto, informatikos fakulteto Informacijos sistemų katedroje pavyzdys. Šiame straipsnyje analizuojamas UML sekų (angl. Sequence) modelio generavimo galimybės pagal pavyzdyje pateiktą veiklos metamodelį.



2 pav. Veiklos metamodelio klasių modelis (2)

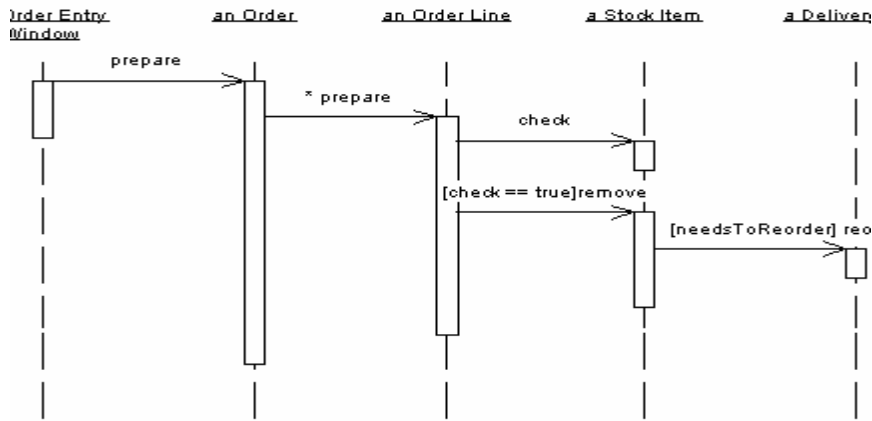
3. UML sąveikos modelis

UML (angl. Unified Modeling Language) – universali modeliavimo kalba. Nurodanti standartinį rinkinį diagramų ir notacijų, skirtų objektiškai orientuotų sistemų modeliavimui. UML aprašo diagramų ir simbolių semantines reikšmes ir yra skirta programinei įrangai specifikuoti, konstruoti, vizualizuoti ir dokumentuoti.(4) UML diagramos skirstomos į dvi pagrindines rūšis: statines ir dinamines. Šiame darbe bus analizuojamos sistemos dinamiką atvaizduojančios sąveikos (angl. Interaction) diagramos dalis sekų (angl. sequence) diagramos.

Į objektų tarpusavio sąveikos sampratą yra įtraukiami du aspektai: objektų tarpusavio komunikacija ir objektų tarpusavio ryšiai. Atitinkamai sąveikos proceso elementams galima išskirti dviejų tipų sąveikos modelius, kuriuos galima sukurti UML modeliavimo kalba: (5)

3. **Bendradarbiavimo (angl. collaboration).** Šioje diagramoje komunikaciniai ryšiai nėra pateikiami realiame laike, todėl komunikacijos eiga yra numeruojama.(9)

4. **Sekos (angl. sequence).** Ši diagrama yra naudojama dekomponuoti sudėtingus sąveikų tarp objektų scenarijus realių laiku.(7)

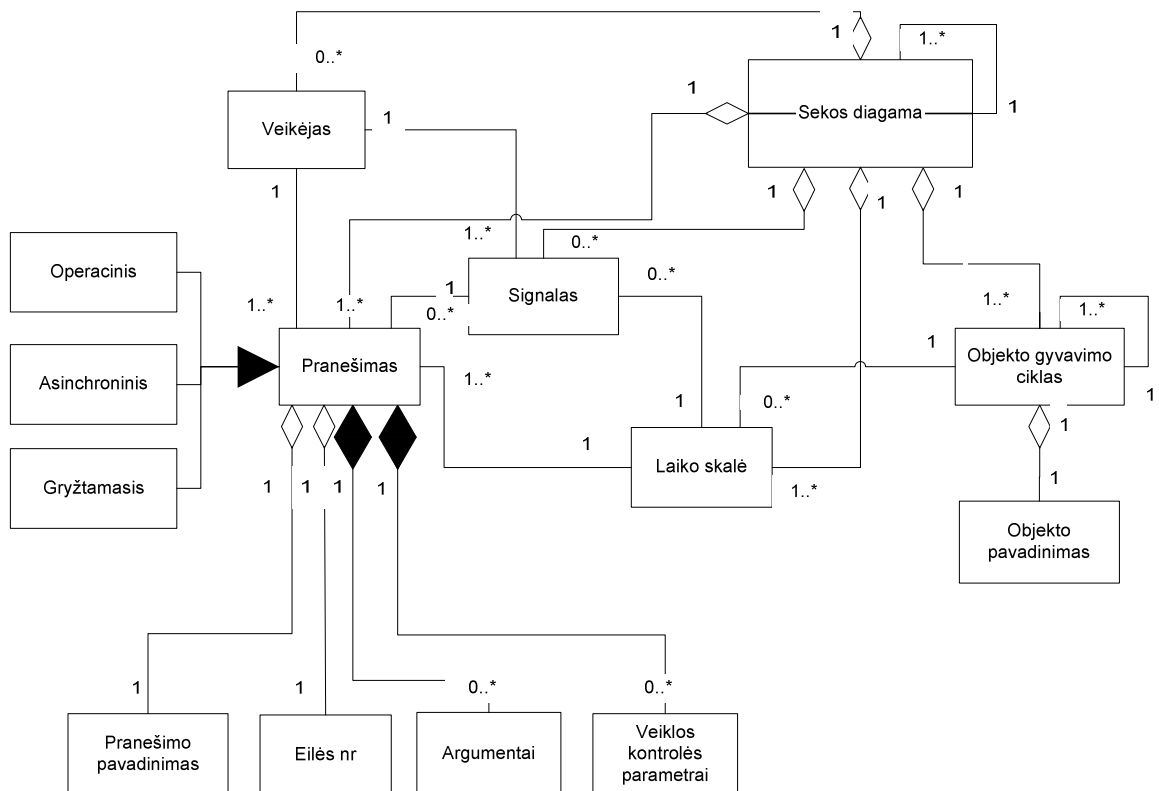


3 pav. Sekos diagramos pavyzdys (8)

Aptarus UML sąveikos modelių, rūšis bus analizuojamas vienos iš sąveikos diagramų sudedamųjų dalių sekų diagramos generavimo procesas, remiantis anksčiau straipsnyje pateiktu veiklos metamodeliu.

4.UML Sekų metamodelio generavimas

4 pav. yra pateikiamas sugeneruotas UML (angl. sequence) metamodelis. Modelyje yra parodoma sekų diagramos sudėtinės struktūrinės dalys jų tipai ir tarpusavio ryšiai.



4pav. UML sekų metamodelis

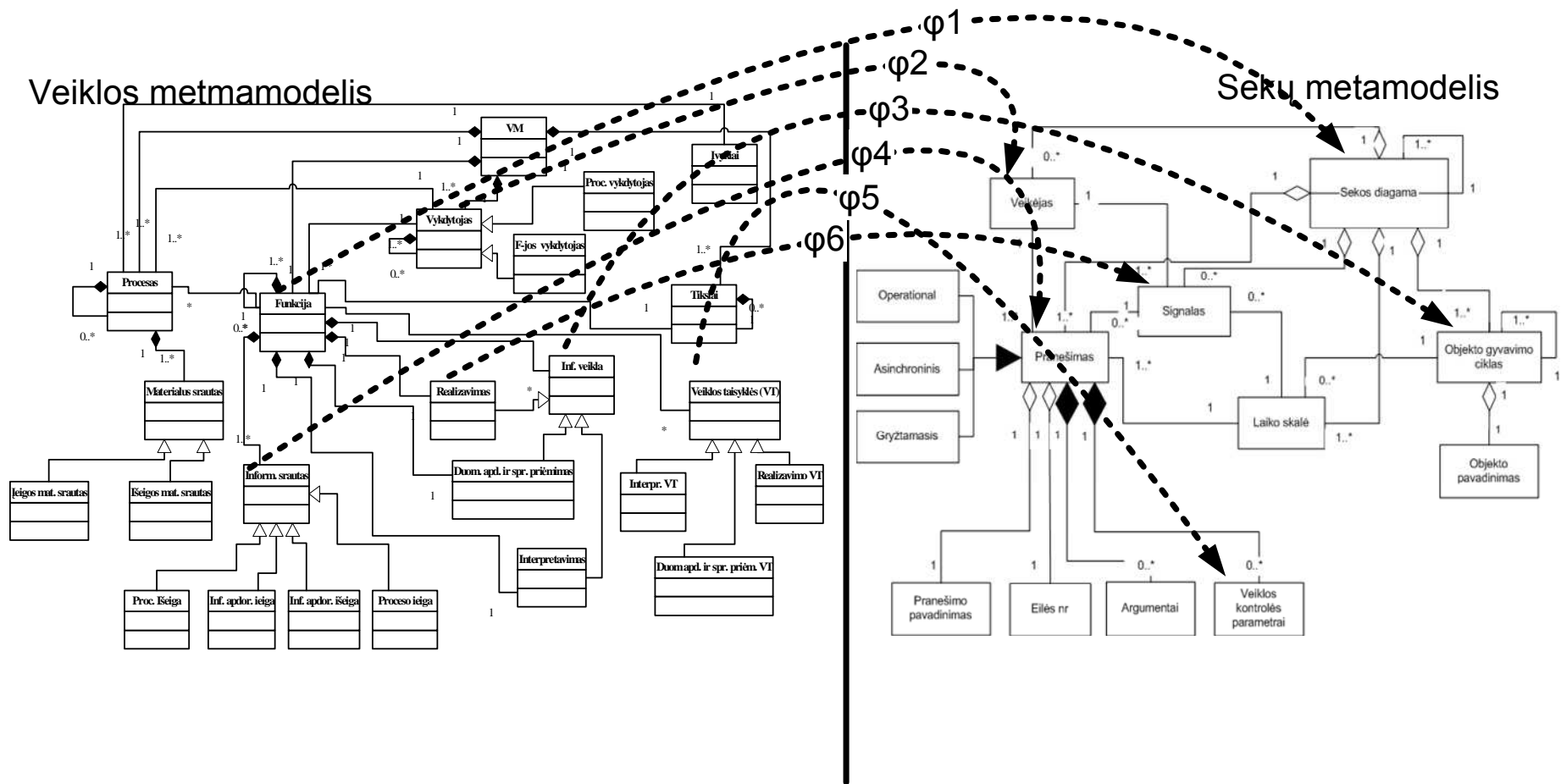
1 lentelėje matome veiklos metamodeliu grindžiamo UML sekų metamodelio generavimo proceso formalų aprašymą susidedantį iš veiklos ir sąveikų metamodelių atitinkamos klasės pavadinimų ir atvaizdžio nurodančio kaip siejasi metamodelių klasės viena su kita.

1 lentelė Veiklos metamodelio klasių atvaizdžiai į sekų metamodelį

Veiklos metamodelio klasės pavadinimas	Atvaizdis (→)	Sąveikos metamodelio klasės pavadinimas
VM.funkcija	$\phi 1$	SM.sekos diagrama
VM.Vykdytojas	$\phi 2$	SM.vykdytojas
VM.informacinė veikla	$\phi 3$	SM.objekto gyvavimo ciklas
VM informacinis srautas	$\phi 4$	SM.pranešimas
VM.veiklos taisyklės	$\phi 5$	SM.veiklos kontrolės parametrai
VM.Realizavimas	$\phi 6$	SM.Signalas

5 paveiksle matome 1 lentelėje formaliai aprašytus atvaizdžius pateiktus grafiškai. Kiekviena įmonėje atliekama funkcija yra aprašoma sekos diagrama $\phi 1$. Funkciją atliekantis vykdytojas atvaizduojamas kaip sekos diagramos vykdytojas sekos metamodelyje dekomponuojamų sekų iniciatorius $\phi 2$. Su funkcija susijusios informacinės veiklos yra atvaizduojamos objekto gyvavimo cikluose sekos metamodelyje $\phi 3$. Informaciniai srautai kurių pagalba atliekama funkcija atitinka pranešimus sekos metamodelyje $\phi 4$. Funkciją kontroliuojančios veiklos taisyklės veiklos metamodelyje atitinka veiklos kontrolės parametrus sekų metamodelyje $\phi 5$. Funkcijos įgyvendinimo sprendimai ir procedūros atvaizduojamos veiklos metamodelyje kaip klasė „realizavimas“, sekų metamodelyje atitinka „signalas“ klasę žymimą $\phi 6$.

5 paveiksle yra grafiškai pateikiamas sąveikos metamodelio generavimo proceso etapas, kurio metu susijusios veiklos metamodelio klasės yra atvaizduojamos sekų metamodelyje.



5pav. Veiklos metamodelio atvaizdavimas į UML sekų metamodelį

IŠVADOS

Išanalizavus veiklos metamodelio ir sąveikos metamodelio sudėtį nustatyta, kad Veiklos metamodelyje saugomų žinių pakanka UML sąveikos modelio generavimui.

Atlikus tyrimą, nustatyta kad sąveikos modelio generavimas intelektualizuoja IS inžinerijos projektavimo etapą tuo, kad UML sąveikų modelis sukuriamas interaktyviai, t.y. vykdant generavimo iš veiklos modelio (kuriame saugomos dalykinės srities žinios yra patikrintos valdymo požiūriu) algoritmą dalyvaujant analitikui ar projektuotojui.

Viena iš priežasčių, kuri apriboja CASE sistemų funkcionalumą (pavyzdžiui yra ribotas projektinių modelių ir programinio kodo generavimo galimybės) yra nepakankamos empiriniu būdu sudarytų veiklos modelių savybės ir nepakankamos vartotojo informacinių reikalavimų specifikacijų savybės. Šią problemą tikslinga spręsti taikant veiklos metamodelį, kuriame saugomos valdymo požiūriu patikrintos dalykinės srities žinios UML modelių (šiuo atveju sekų modelio) generavimui.

Literatūros sąrašas

1. Gudas S. Lopata A., Skersys T. Framework for knowledge based IS Engineering, ADVIS, 2004
2. Gudas S. Lopata A. Veiklos modelių grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikavimo metodas, daktaro laipsnio disertacija, VU KHF 2005
3. Gudas S., Lopata A. Žiniomis grindžiamos informacijos sistemų inžinerijos bruožai, 2005
4. Bock Conrad. UML 2 Activity and Action models// http://www.jot.fm/issues/issue_2003_07/column3/#ref1, 2005 10 30
5. Object Management Group, OMG Unified Modeling Language, version 1.5, //http://www.omg.org/cgi-bin/doc?formal/03-03-01, 2005 11 30 .
6. GERAM:Generalised Enterprise Reference Architecture and Methodology. IFIP–IFAC Task Force on Architectures for Enterprise Integration // <http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html> 2006 03 28
7. Larman C. UML interaction diagrams// <http://www.informit.com/articles/article.asp?p=360441&seqNum=2> , 2005 10 30
8. Interaction diagrams: information of home page for a business process reengineering project done in partial fulfillment of requirements for ISDS 7525 at Louisiana State University <http://isds.bus.lsu.edu/cvoc/learn/bpr/cprojects/spring1998/modeling/interaction.html>, 2005 11 01
9. [Nuno Jardim Nunes, João Falcão e Cunha](http://math.uma.pt/tupis00/submissions/nunes_Cunha/nunescunha.html) Towards a UML profile for interaction design: the Wisdom approach// http://math.uma.pt/tupis00/submissions/nunes_Cunha/nunescunha.html, 2005 10 30
10. European Committee for Standardization. ENV 12 204: Advanced Manufacturing Technology - Systems Architecture - Constructs for Enterprise Modelling, CEN TC 310/WG1, 2006 03 28
11. European Committee for Standardization. ENV 40 003: Computer Integrated Manufacturing - Systems Architecture - Framework for Enterprise Modelling, CEN/CENELEC, 2006.03.28.

Summary:

Š. Čirvinskas, A. Lopata UML sequence model generation based on enterprise model

Paper overviews Knowledge-Based IS Engineering development. The enterprise metamodel is described. Also UML diagram classification is given. The main objective of the paper is the analysis of the generation process of UML sequence model base on enterprise metamodel.