

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

EGLĖ BALEŽENTYTĖ

VERSLO PROCESŲ LOGIKOS VERIFIKAVIMAS
DIRBTINIO INTELEKTO PRIEMONĖMIS

Magistro darbas

Vadovas
doc. R. Misevičienė

KAUNAS, 2014

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

EGLĖ BALEŽENTYTĖ

VERSLO PROCESŲ LOGIKOS VERIFIKAVIMAS
DIRBTINIO INTELEKTO PRIEMONĖMIS

Magistro darbas

Vadovas:
doc. R. Misevičienė

(data, parašas)

Recenzentas:
dr. V. Jakštienė

(data, parašas)

Studentas:
E. Baležentytė

(data, parašas)

KAUNAS, 2014

AUTORIŲ GARANTINIS RAŠTAS
DĖL PATEIKIAMO KŪRINIO

2014 - 05 - 28 d.

Kaunas

Autoriai, _____ Eglė Baležentytė _____,
(vardas, pavardė)

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis magistro darbas (toliau vadinama – Kūrinys) Verslo procesų logikos verifikavimas dirbtinio intelekto priemonėmis

(kūrinio pavadinimas)

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

- 1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
- 2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
- 3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
- 4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
- 5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
- 6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
- 7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autoriai

(vardas, pavardė) (parašas)

(vardas, pavardė) (parašas)

(vardas, pavardė) (parašas)

(vardas, pavardė) (parašas)

SANTRAUKA

Šiuolaikinės įmonės siekia sutrumpinti savo darbų atlikimo laiką, pagerinti darbo principus ir kainą, bei pasiekti didesnę vartotojų pasitenkinimą teikiančiomis paslaugomis, tačiau susiduria su poreikiu pertvarkyti savo veiklos procesus, reorganizuoti veiklą ar ją atnaujinti. Todėl tokie veiksmai skatina naujų verslo modeliavimo priemonių kūrimą. Kuriant tokias verslo modeliavimo priemones susiduriama su tokiais klausimais kaip patikimumas, ekonominis efektyvumas, sistemų projektavimas.

Atlikta mokslinių šaltinių analizė parodė, kad šiuo metu verslo procesų logikai aprašyti naudojami metodai (tokie kaip verslo procesų modeliavimo notacija (BPMN), unifikuota modeliavimo kalba (UML veiklos diagramos) bei kiti formalizmai) yra gana sudėtingi ir, verslo atstovams, kurie nežino minėtų formalių procesų užrašymo būdų, sunku susišnekėti su informacinių technologijų specialistais modeliuojančiais šiuos procesus.

Todėl baigiamojo darbo tikslas buvo pasiūlyti naują verslo procesų analizės metodiką, panaudojant procesų specifikavimui grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose dirbtinio intelekto priemonėmis. Dirbtinio intelekto Prolog kalboje realizuotas išvedimo mechanizmas, pagrįstas paieškos algoritmais, leidžia labai nesudėtingai aprašyti ir analizuoti grafus. Be to, grafų forma gerai suprantama tiek verslo tiek ir informacinių technologijų specialistams.

Darbas susideda iš 4 pagrindinių dalių: Pirmoje dalyje apibrėžtas verslo procesas, išanalizuoti ir palyginti verslo procesų modeliai, pasiūlyta verslo procesų verifikavimo metodika. Antroje dalyje pateikiami verslo procesų verifikavimo metodai. Trečioje dalyje aprašomos siūlomos dirbtinio intelekto priemonės, skirtos verslo procesų verifikavimui, aprašyta sukurta ekspertinė sistema. Ketvirtoje dalyje pateikti eksperimentiniai rezultatai.

Darbo rezultatai buvo pateikti „Kompiuterininkų dienos 2013“ konferencijoje ir publikuoti medžiagoje straipsnyje „Verslo procesų verifikavimas, panaudojant modifikuotą paieškos algoritmą“.

SUMMARY

Modern companies constantly seek to reduce their work installation time, improve working methods while lowering cost and also to achieve greater customer satisfaction by providing services. Unfortunately these companies are faced with the need to transform their business processes to reorganize work. Therefore, such actions encourage new business modeling tools. Developing business modeling tools brings-up issues such as reliability, cost-effectiveness, and system design.

Analyzed research sources show that methods describing current business process logic are quite complex (such as business process modeling notation (BPMN), Unified Modeling Language (UML activity diagrams) and other formalisms). For business representatives who do not know these formal process techniques, it is difficult to understand information technology specialists who are modelling these processes.

Therefore, the goal of this work is to offer a new analysis methodology for business processes using graphs, process specifications and verify the correctness of the search algorithm using graph artificial intelligence tools. Artificial intelligence prolog language realized output mechanism based with search algorithms, makes it very simple to describe and analyze graphs. In addition, form of the graphs is well understood by both business and information technology professionals.

The work consists of four main parts: In the first part business process was described and analyzed. Also business process models were compared, new business process verification methodology was offered. The second part is focused on describing the business process verification method. In the third part offered artificial intelligence tools for business process verification were described. Expert developed system was also described. The fourth section presents experimental results.

The results of this work were presented in the Computer Days 2013 Conference and published in the article “Business process verification using a modified search algorithm”.

TURINYS

Paveikslų turinys.....	7
Lentelių turinys.....	7
ĮVADAS	9
1. VERSLO PROCESŲ LOGIKOS MODELIAVIMO GALIMYBIŲ ANALIZĖ.....	10
1.1. Verslo procesų modeliavimas	10
1.2. Verslo procesų modelių palyginimas	12
1.2.1. Verslo procesų UML veiklos diagrama.....	13
1.2.2. Verslo procesų BPMN diagramos	14
1.2.3. Grafiškai pagrįstas verslo procesų modeliavimas	15
1.3. Grafų analizė	17
1.4. Pirmojo skyriaus išvados.....	18
1.5. Siūloma magistrinio darbo problemos tyrimo metodika.....	18
2. VERSLO PROCESŲ VERIFIKAVIMAS	20
2.1. Pasiekiamumo savybių tyrimas	20
2.2. Formalus verifikavimo uždavinio užrašymas.....	22
2.3. Verslo proceso „Užsakymas gamybai“ verifikavimo pavyzdys	23
2.4. Antrojo skyriaus išvados	24
3. VERIFIKAVIMAS DIRBTINIO INTELEKTO SISTEMOSE	25
3.1. Dirbtinio intelekto taikymo sritys	25
3.2. Ekspertinių žinių vaizdavimas produkcijų sistema	26
3.3. Ekspertinių žinių valdymas loginio programavimo kalba PROLOG.....	29
3.4. Verslo procesų verifikavimo ekspertinė sistema.....	30
3.4.1. Verslo procesų verifikavimo ekspertinės sistemos dedamosios.....	30
3.4.2. Paieška būsenų erdvėje realizuota Prolog kalba.....	31
3.4.3. Prolog sakiniai, tikrinantys aklaviečių ir ciklų nebuvimą	34
3.4.4. Ekspertinės sistemos vartotojo sąsaja.....	34
3.5. Trečiojo skyriaus išvados	37
4. EKSPERIMENTAI	38
4.1. Verslo proceso verifikavimo pavyzdys	38
4.2. Gyvybės draudimo verslo proceso verifikavimas	39
4.3. Internetinės svetainės registracijos proceso verifikavimas	42
4.4. Metinės pajamų/ mokesčių deklaracijos užpildymo proceso verifikavimas	45
4.5. Ketvirtą skyriaus rezultatai ir išvados	50
5. IŠVADOS.....	52
6. LITERATŪROS SĄRAŠAS.....	53
7. PRIEDAI	55
7.1. Moksliniai rezultatai, publikuoti „Kompiuterininkų dienos 2013“ konferencijos medžiagoje	55

Paveikslų turinys

1 pav. Organizacijos struktūra [5]	10
2 pav. Procesinės ir funkcinės organizacijos [8]	10
3 pav. Proceso veikimo schema [8]	11
4 pav. Poterio įmonės vertės grandinės modelis [7]	11
5 pav. Procesų ryšių schema [7]	12
6 pav. Veiklos diagramos pavyzdys [11].....	13
7 pav. Darbuotojo išleidimas į atostogas UML veiklos diagrama [12]	14
8 pav. Darbuotojo išleidimas į atostogas BPMN veiklos diagrama [12]	15
9 pav. Darbuotojų išleidimo į atostogas orientuotas grafas [12]	16
10 pav. Paieška į gylį [17]	17
11 pav. Paieška į plotį [18]	18
12 pav. Grafo atvaizdavimas į medį [19]	18
13 pav. Verslo procesų verifikavimo metodika.....	19
14 pav. Verslo diagramų verifikavimas [16]	20
15 pav. Pasiekiamumo iliustravimas	21
16 pav. Modifikuotas paieškos algoritmas	21
17 pav. Grafo pavyzdys [9]	23
18 pav. Verslo proceso pavyzdys naudojantis BPMN notaciją [13]	23
19 pav. BPMN diagramos procesas atvaizduotas grafu	24
20 pav. Paieškos procesas atvaizduotas medžiu	24
21 pav. Taisyklėmis pagrįsta ES struktūra [27].....	26
22 pav. Išvedimas produkcijų sistemoje.....	28
23 pav. Verslo procesų verifikavimo sistemos struktūra.....	30
24 pav. Paieškos uždavinio užrašymas Prolog kalba [32].....	31
25 pav. Paieškos gilyn grafinis vaizdavimas [32]	32
26 pav. Gyvybės draudimo UML veiklos diagrama [9].....	33
27 pav. Verslo proceso grafas.....	33
28 pav. Verslo proceso sprendimo medis.....	34
29 pav. Vartotojo sąsajos INTERF predikatai.....	35
30 pav. Grafo pavyzdys [9]	38
31 pav. Ekspertinės sistemos Prolog rezultatai	39
32 pav. Gyvybės draudimo UML veiklos diagrama [9].....	40
33 pav. Gyvybės draudimo proceso grafas.....	40
34 pav. Gyvybės draudimo proceso realizavimas su Prolog kalba	41
35 pav. Gyvybės draudimo proceso sprendimo medis	41
36 pav. Internetinės svetainės prisijungimo proceso veiklos diagrama [16] [33]	43
37 pav. Internetinės svetainės veikimo grafas	44
38 pav. Internetinės svetainės prisijungimo realizavimas su Prolog	45
39 pav. Metinės pajamų mokesčio deklaracijos grafas	48
40 pav. Metinės pajamų mokesčio deklaracijos verifikavimas su Prolog.....	49
41 pav. Metinės pajamų mokesčio deklaracijos verifikavimas su Prolog (tęsinys)	49
42 pav. Komandos <i>interf_pagalba</i> rezultatai	50
43 pav. Komandos <i>užkrauti</i> rezultatai	50

Lentelių turinys

1 lentelė. UML, BPMN, grafų pagrindiniai žymėjimai.....	16
2 lentelė. Metinės pajamų deklaracijos formos GPM 308 formos laukai	46

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Verslo procesas - yra veikla arba veiklos, kurios bus atliktos siekiant tam tikro organizacijos tikslo.

Verslo proceso modeliavimas - sukūrimas abstraktaus arba materialaus objekto kurį tiriant, sužinoma tam tikra informacija apie realų verslo procesą, t. y. modelio originalą arba prototipą.

Verslo proceso veikla - tam tikra užduotis (vienas loginis žingsnis), įvykdoma proceso eigoje.

Dirbtinis intelektas (*anlg. artificial intelligence*) - programinė sistema, imituojanti žmogaus mąstymą kompiuteryje.

Ekspertinė sistema tai dirbtinio intelekto sistema, sukurta konkrečios probleminės srities uždavinių sprendimui.

BPMN (*anlg. Business Process Modeling Notation*) - grafinė verslo procesų specifikavimo notacija.

BPD (*anlg. Business Process Diagram*) - verslo proceso diagrama, naudojama *BPMN* notacijoje.

UML (*anlg. Unified Modeling Language*) – kalba, skirta specifikuoti, vizualizuoti ir dokumentuoti programinės įrangos sistemas, taip pat naudojama ir verslo modeliavimui ir kitokioms ne programinėms sistemoms.

PERT (*anlg. Program Evaluation and Review Technique*) yra statistinis įrankis naudojamas projektų valdymui, kuris analizuoja ir reprezentuoja užduotis, kurios yra skirtos projekto užbaigimui.

IDEF (*anlg. Integration Definition Function Modeling*) - ICAM aprašymų šeimų kalba, kuri yra procesų modeliavimo ir funkcionavimo vaizdavimo standartų šeima, naudojama techninėms sistemoms ar verslo procesams kurti ar gerinti.

Validavimas - veikla, kuria siekiama įsitikinti, kad sistema atitinka reikalavimus (atsako į klausimą, ar kuriamas teisingas produktas?).

Verifikavimas - tai veikla, kuria siekiama įsitikinti, kad tam tikra funkcija vykdoma korektiškai, ar sistema atitinka keliamus reikalavimus.

ĮVADAS

Vis daugiau įmonių, siekdamas sutrumpinti savo darbų atlikimo laiką, pagerinti darbo principus ir kainą, bei pasiekti didesnę vartotojų pasitenkinimą teikiančiomis paslaugomis, susiduria su poreikiu pertvarkyti savo veiklos procesus, reorganizuoti veiklą ar ją atnaujinti. Informacinių technologijų vystymasis ir jų svarba šiuolaikiniame versle skatina naujų verslo modeliavimo priemonių kūrimą. Tuo pačiu susiduriama su patikimumo, ekonominio efektyvumo, sistemų projektavimo klausimais.

Laikui bėgant pasaulinėje rinkoje atsirado nemažai produktų užsiimančių verslo procesų modeliavimu, valdymu, pavaizdavimu bei vykdymu. Tokiomis aplinkybėmis susiformavo naujos problemos, susijusios su pastarųjų produktų ir įvairių modelių suderinamumu.

Kompiuterizuojant verslo procesus bei kuriant darbų srautų valdymo sistemas, pirmiausia reikia apibrėžti verslo proceso logiką. Šis aprašas turi būti tikslus ir korektiškas. Sudarant, t.y. formalizuojant gana sudėtingą verslo procesų valdymo logiką, juos būtina verifikuoti, t. y. įsitikinti ar nėra nurodyta neteisingų veiksmų. Dėl to valdymo algoritmus reikia formalizuoti.

Šiuo metu formaliai verslo procesų logikai aprašyti naudojami gana įvairūs metodai. Dažniausiai naudojama verslo procesų modeliavimo notacija (BPMN), unifikuota modeliavimo kalba (UML veiklos diagramos), Petri tinklai, verslo taisyklės bei kitos notacijos. Jų privalumas toks, kad jos leidžia grafiškai dokumentuoti sudėtingą procesų logiką, tačiau, norint patikrinti šių notacijų teisingumą reikalingos sudėtingos programinės priemonės [1][2][3][4]. Be to, verslo atstovams, kurie nežino minėtų formalių procesų užrašymo būdų, sunku susišnekėti su informacinių technologijų specialistais modeliuojančiais šiuos procesus. Todėl ieškoma vis geresnių būdų dalykinei veiklos sričiai aprašyti grafiniais modeliais, suprantamais visiems.

Darbe siūloma verslo procesų logiką vaizduoti grafais. Jų privalumas toks, kad jie leidžia grafiškai aprašyti sudėtingą verslo procesų logiką. Grafų forma gerai suprantama tiek verslo tiek ir informacinių technologijų specialistams. Grafų teisingumo tyrimui siūloma panaudoti grafų paieškos algoritmus panaudojant dirbtinio intelekto priemones. Tam siūloma sukurti ekspertinę sistemą realizuotą Prolog kalba. Dirbtinio intelekto Prolog kalboje realizuotas išvedimo mechanizmas, pagrįstas paieškos algoritmais, leidžia labai nesudėtingai aprašyti ir analizuoti grafus.

Tyrimo sritis – verslo procesų darbų eigos modeliavimo ir analizės problemos, susijusios su verslo procesų specifikavimu, modeliavimu bei analize reikalavimų specifikavimo etape.

Darbo tikslas – pasiūlyti naują verslo procesų analizės metodiką, panaudojant procesų specifikavimui grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose dirbtinio intelekto priemonėmis.

Darbo uždaviniai:

- Išanalizuoti verslo procesų specifikavimo galimybes;
- Išanalizuoti verslo procesų analizės (verifikavimo) priemones;
- Pasiūlyti naują verslo procesų analizės metodiką, kurioje bus sujungti paieškos algoritmai grafuose ir dirbtinio intelekto priemonės;
- Pasiūlytą metodiką pritaikyti realioms uždaviniamis spręsti;

Darbo naujumas – skirtingų trijų sričių apjungimas verslo procesų analizei:

1. Verslo procesų modeliavimas;
2. Paieškos algoritmų taikymas grafuose verslo procesų teisingumo analizei;
3. Dirbtinio intelekto priemonių (ekspertinės sistemos ir programavimo kalbos Prolog) panaudojimas paieškos algoritmų realizavimui.

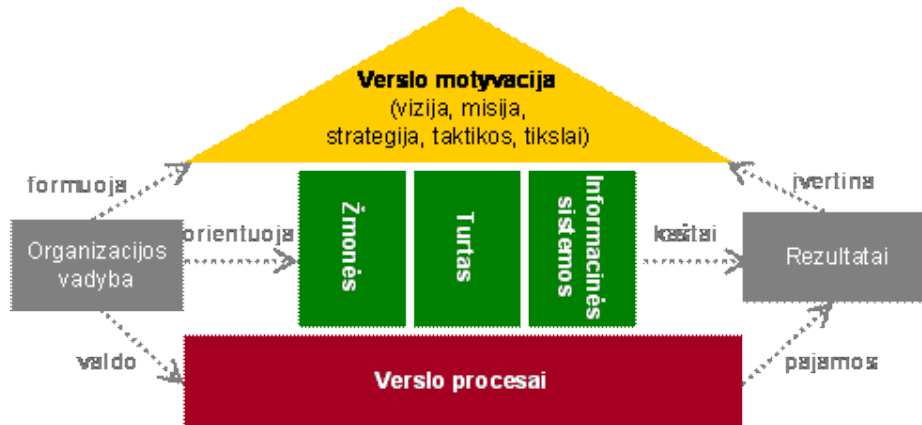
Darbo struktūra. Darbas susideda iš 4 pagrindinių dalių: Pirmoje dalyje apibrėžtas verslo procesas, išanalizuoti ir palyginti verslo procesų modeliai, pasiūlyta verslo procesų verifikavimo metodika. Antroje dalyje pateikiami verslo procesų verifikavimo metodai. Trečioje dalyje aprašomos siūlomos dirbtinio intelekto priemonės, skirtos verslo procesų verifikavimui, aprašyta sukurta ekspertinė sistema. Ketvirtoje dalyje pateikti eksperimentiniai rezultatai.

Darbo rezultatai buvo pateikti „Kompiuterininkų dienos 2013“ konferencijoje ir publikuoti medžiagoje straipsnyje „Verslo procesų verifikavimas, panaudojant modifikuotą paieškos algoritmą“.

1. VERSLO PROCESŲ LOGIKOS MODELIAVIMO GALIMYBIŲ ANALIZĖ

1.1. Verslo procesų modeliavimas

Remiantis [5] žinoma, kad dauguma dabartinių organizacijų turi susikūrę savo vidinę verslo architektūrą (1 pav.), kurių pagrindiniai elementai: verslo motyvacijos modelis (taktika, vizija, tikslai, misija, strategija) ir organizacijos vykdomi verslo procesai, kurie naudoja įvairius resursus (žmones, informacines sistemas, turta). Todėl šie verslo procesai leidžia pasiekti tikslus, nustatytus verslo motyvacijos modelyje.

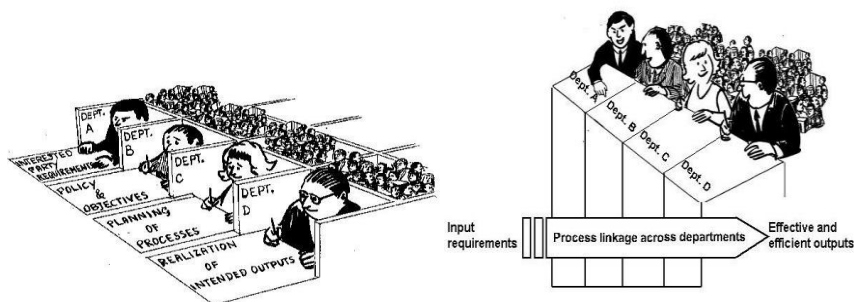


1 pav. Organizacijos struktūra [5]

Sakoma, kad nuo XIX a. iki šių dienų dažniausiai pasitaikantis organizacijos valdymo būdas yra hierarchinis, kai apibrėžiamos darbuotojo funkcijos ir pavaldumas [6]. Teigiama [7], kad dauguma įmonių yra valdomos pagal funkcijas - gamybą, marketingą, projektavimą, finansus, logistiką, pardavimus. Šie veiksmai leidžia išnaudoti specializacijos teikiamus privalumus, nes kiekvienas padalinys specializuojasi atlikti tam tikras funkcijas.

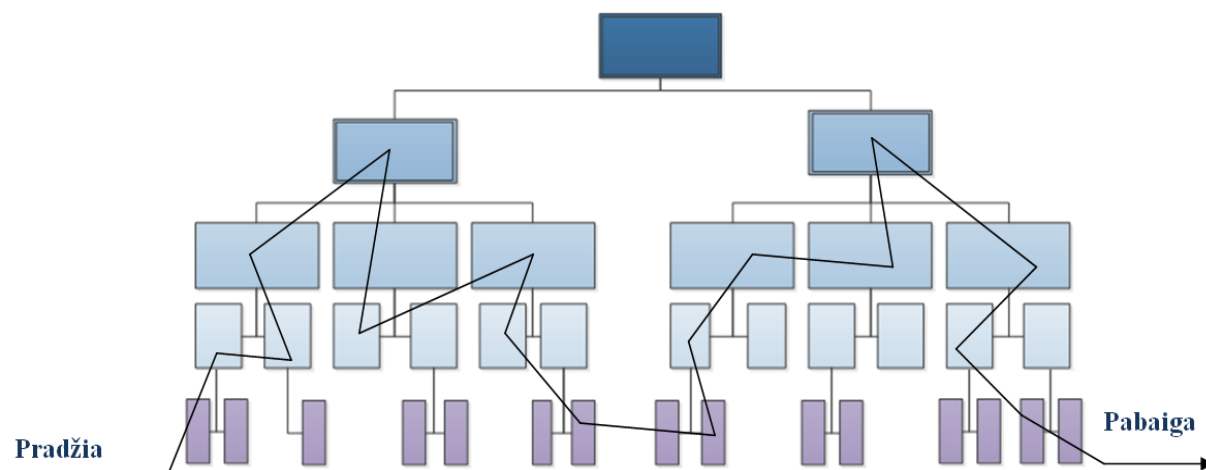
Tačiau įmonės tikslas nėra vien tik geras marketingas, gerai organizuotas pardavimas ar produktų pristatymas atskirai. Svarbiausia yra tarpusavyje susijusių veiksmų seka, kurie veda prie pagrindinio įmonės tikslo.

Dabartinių verslo procesų valdymo tikslai yra perorientuoti organizacijas nuo tradicinio funkcinio požiūrio (žmogui ieškomas darbas), prie procesinio požiūrio (darbui ieškomas žmogus) (2 pav.). Procesinis požiūris - procesai yra pagrindinės priemonės, kuriomis organizacijos siekia atlikti didžiąją dalį veiklos, funkcinis požiūris: organizacija yra sudaryta iš atitinkamų padalinių [8].



2 pav. Procesinės ir funkcinės organizacijos [8]

Darbas įmonėse nejuda aukštyn ir žemyn per organizacijos struktūrą, bet juda išilgai per organizaciją, kaip verslo procesų rinkinys (3 pav.)



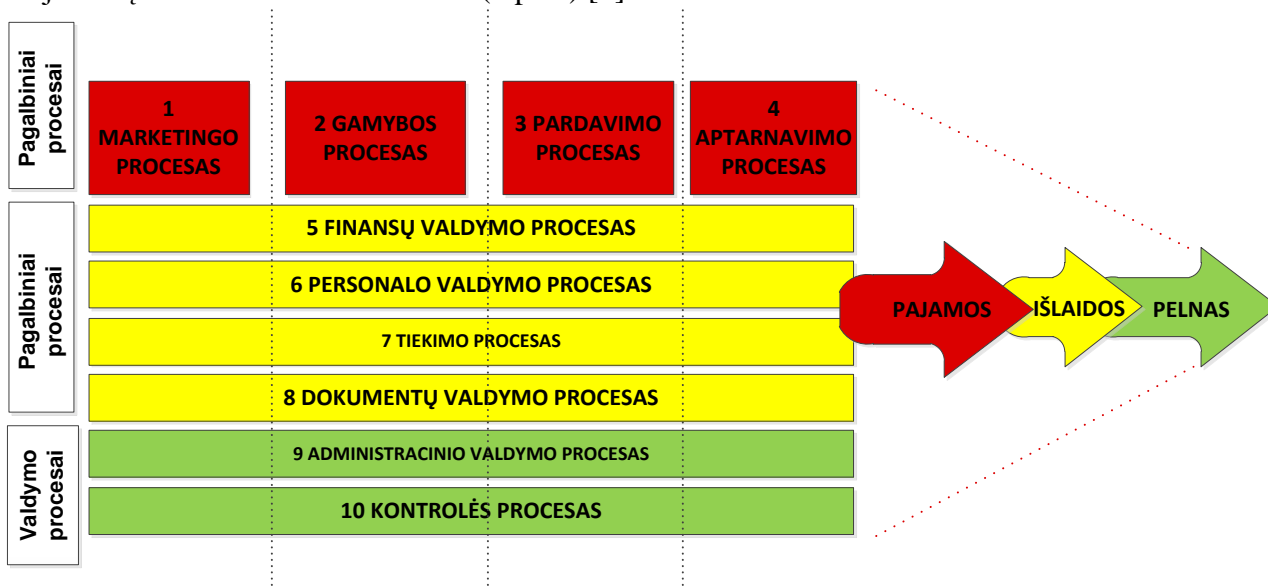
3 pav. Proceso veikimo schema [8]

Įmonės valdymo pagrindas yra darbai ir veiksmai, kurie apjungiami į visumą, o ne vien tik funkcijos. Procesą sudaro tarpusavyje sąveikaujančių funkcijų visuma, kuri pirminius duomenis paverčia rezultatais. Įmonės procesų identifikavimą ir jų valdymo svarbą identifikuoja procesinis požiūris. Procesinis požiūris žiūri naujai į įmonės valdymą, kas skatina iš esmės permąstyti įmonės organizacinę struktūrą, valdymo, kontrolės mechanizmus, efektyvumo rodiklius. Iš esmės šiuolaikinė vadyba remiasi procesais ir jų efektyviu išnaudojimu – detaliu kiekvieno proceso planavimu, organizavimu, kontroliavimu, bei veikiančių procesų tobulinimu. Verslo valdymo sistemos sukuriamos dviem būdais [7]:

- remiantis įmonės hierarchine struktūra;
- remiantis įmonės procesais.

Verslo valdymo sistemos gali būti projektuojamos pagal įmonės vykdomus procesus arba hierarchinę struktūrą. Sparčiai keičiantis rinkos reikalavimams, greitai kinta įmonių struktūros, kas lemia prastą informacinės sistemos ir sukurtų taisyklių efektyvumą šiuo požiūriu. Tad projektavimas pagal procesus pažangesnis ir efektyvesnis. Dėl lėtesnės procesų kaitos įmonėse, projektavimas paremtas jais kur kas efektyvesnis. Tobulėjant vykdomiems procesams, tobulinti esamą informacinę sistemą, procedūras ir taisykles yra daug lengviau, nei kurti naują sistemą pagal hierarchiją [7].

Procesinį požiūrį geriausiai atvaizduoja vieninga procesų schema, kitaip vadinama „procesų žemėlapiu“. Šiam žemėlapiui sukurti naudojamas „Poterio“ įmonės vertės grandinės modelis ir analizuojamos įmonės struktūrinė schema (4 pav.) [7].

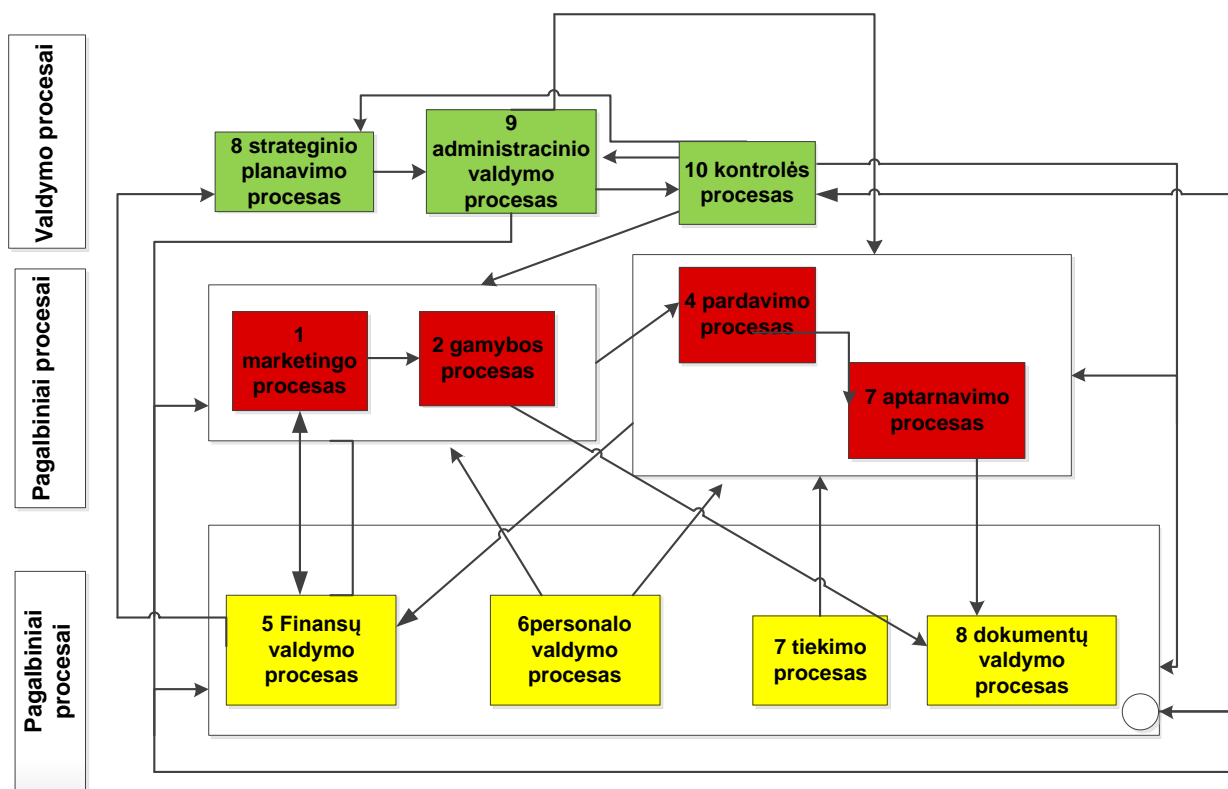


4 pav. Poterio įmonės vertės grandinės modelis [7]

Poterio įmonės vertės grandinės modelį sudaro trys pagrindinės procesų grupės (5 pav.) [7]:

- valdymo procesai – yra naudojami ir susieti su resursų valdymu. Tai yra : procesų gerinimas, strateginis planavimas, kontrolė;
- pagalbiniai procesai - yra tiesiogiai susiję su kuriu produktu pvz.: produkto gamyba, pardavimas, projektavimas ar aptarnavimas.
- aptarnaujantys procesai - yra naudojami aptarnauti pagrindinius bei valdymo procesus.

Verslo procesų optimizavimas bei procesų analizės pagalba galima atpažinti neefektyvias procesų vietas.



5 pav. Procesų ryšių schema [7]

Procesų ryšių schema yra naudojama kaip aiški vaizdinė priemonė, kurios pagalba yra matoma, kaip informacija juda ir apsiekiama įmonės modelyje. Todėl procesų ryšių schema leidžia sukurti įmonės komunikacijos strategiją. Procesų ryšių schemą bei Poterio įmonės vertės grandinės modelis (procesų žemėlapis) yra paprastai sukuriama etapais [7]:

- Numatomi ir nustatomi pagrindiniai bei pagalbiniai procesai;
- Suprojektuojami procesų ryšių schema ir procesų žemėlapių projektai, kurie yra aptariami su įmonės vadovais ir jų darbuotojais;
- Suprojektuotos sistemos yra testuojamos, rastos klaidos yra koreguojamos ir pritaikomos kasdieniniam naudojimui.

Neretai organizacijoje nėra aiškiai suprantama, kokie yra verslo procesai, kaip juos vykdyti. Todėl šie veiksniai labai riboja organizacijų gebėjimus užtikrinti paslaugų ir gaminių kokybę, analizuoti procesų vykdymo rezultatus ir kaštus, sukauptais analizės duomenimis, optimizuoti veiklos procesus, kurių tikslas yra siekti geresnių rezultatų [5].

1.2. Verslo procesų modelių palyginimas

Verslo procesų apibrėžimų yra įvairių, tokių kaip [2]:

- verslo procesas - yra veikla arba veiklos, kurias siekiama įgyvendinti, kad būtų pasiektas tam tikro organizacijos tikslo;

- nuosekliai susijusios veiklos arba užduočių serijos, tokios kaip planavimas, gamyba, pardavimas, kurios atliekamos kartu ir numatant apibrėžtą rezultatų rinkinį.

Dauguma organizacijų aprašo verslo procesus, naudodamos tam nepritaikytus programinės įrangos įrankius – tekstinių redaktorių Microsoft Word, laisvos formos diagramų redaktorių Microsoft Visio, prezentacijų redaktorių Microsoft PowerPoint ar atitinkamas programas iš Open Office programinės įrangos paketo. Bet kuriant ir diegiant būsimus verslo procesus, tokius aprašymus yra labai sudėtinga palaikyti, kadangi informacija yra netiksli, restruktūrizuota, nėra semantinių informacijos ryšių [9].

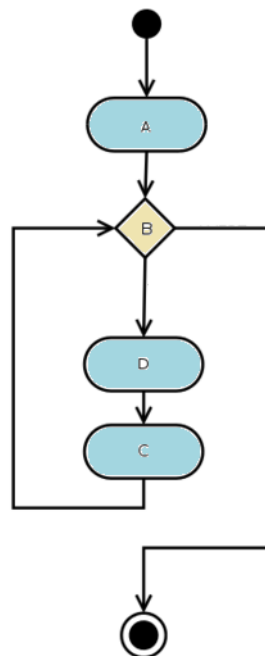
Modelio verslo proceso metodai, pavyzdžiui, srautų diagrama, funkcinė srauto blokinė schema, valdymo srautų diagrama, *Ganto diagrama*, PERT (*angl. Program Evaluation and Review Technique*) diagramos atsirado XX-ojo amžiaus pradžioje. Tarp pirmųjų buvo *Ganto diagramos*, funkcinė srautų schema ir *PERT*, duomenų srautų ir *IDEF* diagramos. Tarp šiuolaikinių metodų yra *Unified Modeling Language* ir verslo procesų modeliavimo žymėjimas bei kitos. Siekiant įgyvendinti verslo procesų analizę, reikėtų naudoti specializuotus modelius [9].

1.2.1. Verslo procesų UML veiklos diagrama

Verslo procesų veiklos diagramos (*angl. Unified Modeling Language (UML)*) yra plačiausiai ir labiausiai žinoma grafinių notacijų vaizdavimo rinkinys - kalba, kuri reikalauja griežtų atvaizdavimo taisyklių. Šios diagramos naudojamos verslo procesų darbų sekoms pavaizduoti, siekdamas nuoseklumo ir aiškaus supratimo, kas kaip turėtų judėti [10].

Kiekvienos *UML* diagramos paskirtis yra pavaizduoti, aprašyti tam tikrus procesus ir veiksmus, kuriuos sudaro tam atitinkami objektai. Nuoseklumas ir konkretumas leidžia vaizdžiai parodyti visą sistemos veiklą, galimybes kurias turi atlikti tam tikra sistema arba tik dalis tos sistemos [10].

UML diagramų yra daug, bet mes nagrinėsime tik tas *UML* diagramas kurios yra susietos ir naudojamos veiklos modeliams. Veiklų diagramomis pavaizduojami sistemoje esantys veiksmi ir veiklos. O modeliavime veiklų diagramas laisvai galima naudoti verslo procesams, kurias sudaro veiksmi ir tų veiksmų srautai. Paveiksle (6 pav.) yra pateiktas elementarus *UML* veiklos diagramos pavyzdys.



6 pav. Veiklos diagramos pavyzdys [11]

Analizuokime pavyzdį, kuriame aprašomas gamybos darbuotojo išėjimo į atostogas procesas, kuris vykdomas dokumentų valdymo sistemoje. Šis procesas susideda iš šešių veiklų ir dviejų sąlygų. Vykdomo proceso valdymą atlieka sistema, kur darbuotojas turi tik inicijuoti procesą arba sukurti dokumentą, kuris inicijuos procesą [12]:

i. Prisijungęs prie sistemos darbuotojas parašo atostogų prašymo dokumentą. Užpildomi pateikti formos laukai: apmokamos/neapmokamos, eilės tvarka/skubos tvarka, atostogų data, komentaras (jei atostogos turi būti suteiktos skubos tvarka). Užpildžius laukus sistema automatiškai patikrina, ar informacija užpildyta korektiškai bei ar norimų atostogų dienų skaičius neviršija leistino skaičiaus.

ii. Gamybos vadovas gauna pranešimą, kad patvirtintų atostogų dokumentą. Peržiūrėjęs prašymą vadovas patvirtina arba nepatvirtina darbuotojo prašymą. Jeigu prašymas yra nepatvirtintas, gamybos vadovas tai praneša darbuotojui. Bet jei prašymą patvirtina, tuomet darbuotojo prašymas nusiunčiamas vadovui (direktoriui).

iii. Kai prašymas nepatvirtintas, tuomet darbuotojas koreguoja prašymą, atsižvelgdamas į gamybos vadovo pastabas (užduotis pataisyti prašymą).

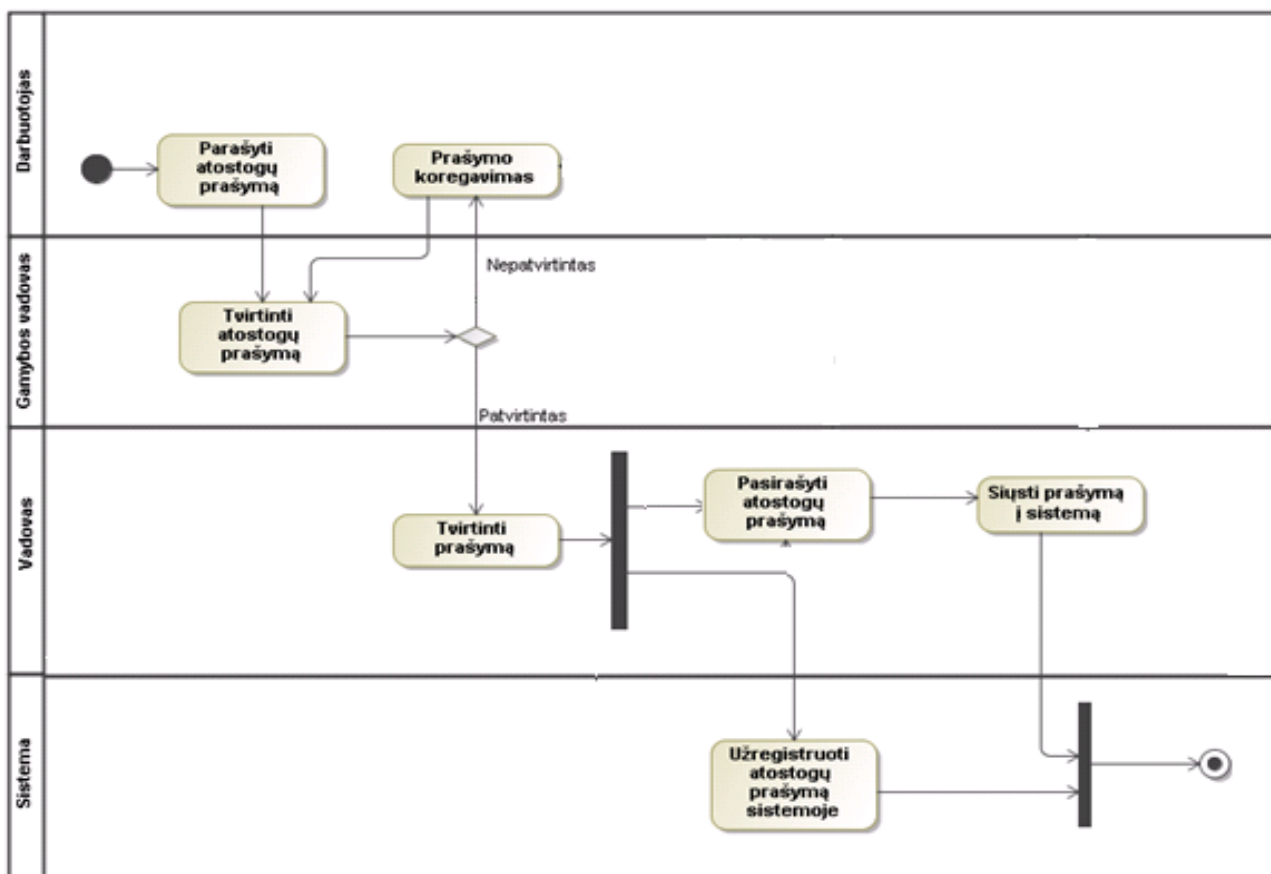
iv. Prašymo nepatvirtinimo atveju situacija analogiška aukščiau išvardintajai.

v. Vadovas patvirtina darbuotojo prašymą. Užregistruojamas prašymas sistemoje.

vi. Prašymo patvirtinimo atveju sistema automatiškai užregistruoja dokumentą. Direktorius gautą dokumentą pasirašo jį skaitmeniniu parašu.

vii. Procesas užbaigiamas.

Aprašyto pavyzdžio veiklos diagrama pateikta (7 pav.) paveiksle [12].



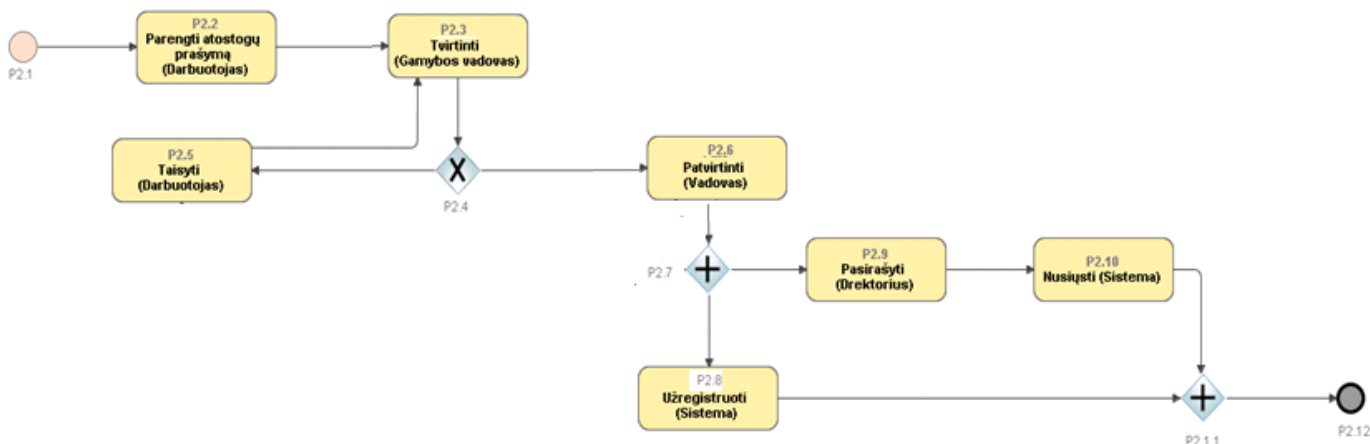
7 pav. Darbuotojo išleidimas į atostogas UML veiklos diagrama [12]

1.2.2. Verslo procesų BPMN diagramos

Kita plačiai naudojama verslo procesų modeliavimo kalba yra BPMN. BPMN pirmoji standarto versija buvo išleista ~ 2004 metais, BPMI.org organizacijos dėka, o dabar šį standarto tobulinimą perėmė organizacija - OMG, kuri yra sukūrusi UML standarto veiklos diagramą [12], [13].

BPMN yra sistema grafinių žymėjimų, kuri skirta verslo procesų schemos pavaizduoti. Verslo procesų diagrama BPMN kuriamos panašiais principais kaip ir tradicinės blokų schemos [14], [2].

BPMN diagrama, taip kaip ir UML veiklos diagrama, vaizduoja grafais grįstus modelius. Žiūrint formaliai, BPMN yra tam tikras grafas su įvestomis konstravimo taisyklėmis. Paveiksle (8 pav.) yra pateikta BPMN pavyzdžio diagrama, kurioje yra pavaizduotas anksčiau nagrinėtas darbuotojų išleidimo į atostogas procesas. Kaip matome, jis beveik niekuo nesiskiria nuo veiklos diagramos, pateiktos 7-tame paveiksle. Vienas pagrindinis skirtumas yra lygiagretumo žymėjimas. BPMN diagramoje yra panaudoti AND tipo vartai. AND tipo vartai reiškia, kad nuo išsišakojimo mazgo lygiagrečiai vykdomos visos išsišakojimo atšakos, o susijungimo mazge laukiama, kol visos įeinančios šakos bus įvykdytos.



8 pav. Darbuotojo išleidimas į atostogas BPMN veiklos diagrama [12]

1.2.3. Grafais pagrįstas verslo procesų modeliavimas

Verslo procesus taip pat galima pavaizduoti ir grafais. Kadangi grafai leidžia grafiškai aprašyti nesudėtingą procesų logiką, o tokia grafų forma gerai suprantama tiek verslo atstovams, tiek ir informacinių technologijų specialistams. Toliau trumpai aprašysime grafus.

Grafas $G = (X, A)$ plokštumoje dažnai vaizduojamas atitinkama viršūnių ir jas jungiančių linijų visuma, tačiau praktiškai, grafas užrašomas ir algebrine forma. Paprasčiausia grafo užrašymo forma yra grafo viršūnių ir lankų išvardijimas [15].

Grafo $G = (X, A)$ viršūnių aibė: $X = \{x_1; x_2; \dots; x_n\}$, lankų aibė: $A = \{a_1; a_2; \dots; a_n\}$.

Užrašius grafą tokia forma, vis tiek neaišku, kurias viršūnes jungia lankai, t.y. vis tiek reikia nusakyti grafo struktūrą – esamus ryšius. Tuomet kiekvieną lanką galima aprašyti, nurodant jo pradinę ir galinę viršūnes. Grafo lankus galima žymėti iš karto dviem indeksais, rodančiais lanko pradinę ir galinę viršūnes [15].

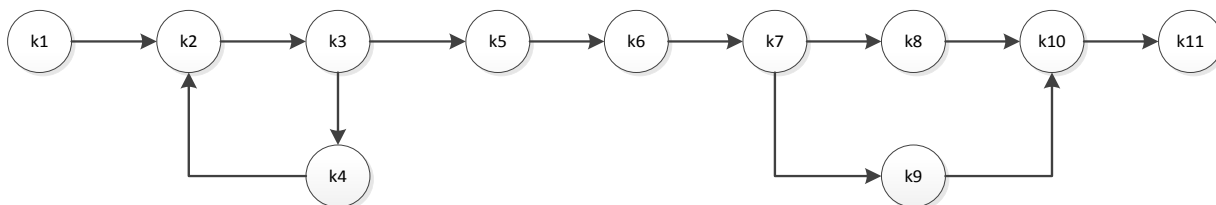
Praktiškai grafo lankai vaizduojami atvaizdu. Atvaizdas $\Gamma(x_i)$ nurodo tas grafo viršūnes, kurios pasiekiamos iš viršūnės x_i . Tokios užrašymo formos grafas žymimas $G = (X, \Gamma)$.

Kadangi atvaizdas $\Gamma(x_i)$ rodo tas viršūnes $x_j \in X$, kurioms grafe $G = (X, \Gamma)$ egzistuoja lankas (x_i, x_j) , tai atvirkštinis atvaizdas $\Gamma^{-1}(x_k)$ rodys tas viršūnes $x_k \in X$, kurioms grafe $G = (X, \Gamma)$ egzistuoja lankas (x_k, x_i) .

Dvi grafo viršūnės, kurios sujungtos bent vienu lanku, vadinamos *gretimomis* arba *kaimyninėmis*, priešingu atveju jos vadinamos nepriklausomomis.

Viršūnių seka $p = \{v_{i0}, v_{i1}, \dots\}$ yra vadinama *k - keliu* (maršrutu, angl. *path*), jei sekos visos gretimos viršūnės yra sujungtos lankais [15],[16].

Anksčiau aprašytas darbuotojų išleidimo procesas gali būti atvaizduotas ir grafu, kuris parodytas paveiksle (9 pav.).

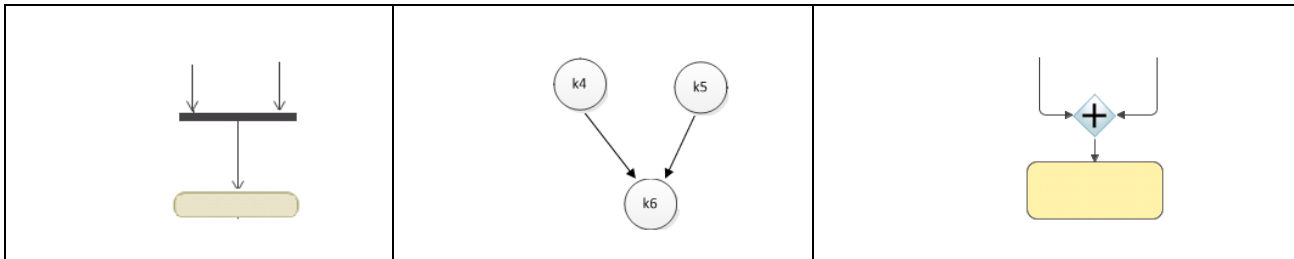


9 pav. Darbuotojų išleidimo į atostogas orientuotas grafas [12]

1 lentelėje yra pateikti pagrindiniai nagrinėtų verslo procesų žymėjimai. Pateikiama UML, BPMN [1] ir grafų žymėjimo simboliai, kurie yra naudojami veiklos diagramoms vaizduoti.

1 lentelė. UML, BPMN, grafų pagrindiniai žymėjimai

UML diagrama	Grafas	BPMN diagrama
Pradžios mazgas (angl. Initial node)		
Diagramos pabaigos mazgas (angl. Activity final node)		
<p><i>Veiksmas (angl. Activity)</i> – tai diagramoje vaizduojami veiksmai: vykdomi, arba tik įvyksiantis. Vaizduoja vieną veiksmą diagramoje, kuris būtinas, kad įvyktų sekantis žingsnis</p>		
<p><i>Sprendimas (angl. Decision point)</i> – rombas veiksmas turintis pasirinkimą (šios sąlygos yra išeinantys srautai, kurie leidžia pasirinkti sekančius veiksmus)</p>		
<p><i>Išsišakojimas (angl. Fork)</i> - įeinantis vienas srautas iš kurio išeina du kiti srautai</p>		
<p><i>Sujungimas (angl. Join)</i> - kelis įeinančius srautus į vieną išeinantį</p>		



1.3. Grafų analizė

Grafų analizė – tai kelio (arba kelių) paieška grafo viršūnių aibėje. Daugelio grafų teorijos uždavinių sprendimo algoritmų pagrindą sudaro sisteminga grafo viršūnių peržiūra, t.y. toks grafo viršūnių apėjimas, kad kiekviena viršūnė nagrinėjama vienintelį kartą.

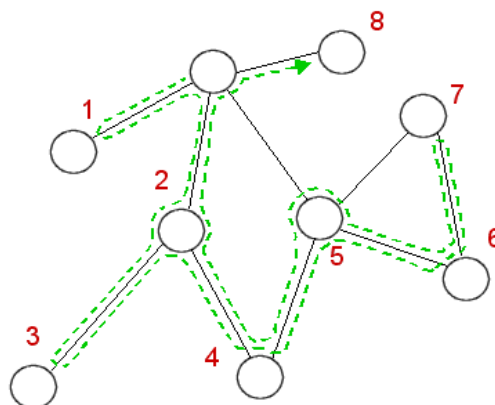
Kelio paieškos problema gali būti aprašyta tokiais komponentais:

- Pradinė būsena. Pradinė būsena nurodo paieškos pradžios viršūnę.
- Galimų perėjimų aibė. Tai visi perėjimai į įpėdinio būsenas.
- Būsenų erdvė. Tai visų galimų būsenų aibė. Pradinė būsena kartu su galimų perėjimų sąrašu apibrėžia kelio paieškos būsenų erdvę.
- Kelias būsenų erdvėje apibrėžiamas kaip būsenų seka, sujungta galimų perėjimų seka.
- Tikslo arba galinė būsena. Apibrėžia siekiamą tikslo būseną.

Visi išvardinti komponentai apibrėžia kelio paieškos problemą. Paieškos idėja labai paprasta. Turint grafą, pradinių viršūnių aibę ir galinių viršūnių aibę, nuosekliai patikrinti kelius, kurie veda iš pradinės į galinę viršūnę. Problemą išsprendus, gaunamas rezultatas, kuris yra kelias nuo pradinės iki galinės būsenos. Taip analizuojant grafą, jo korektiškumą galima apibrėžti tokiais pagrindiniais kriterijais:

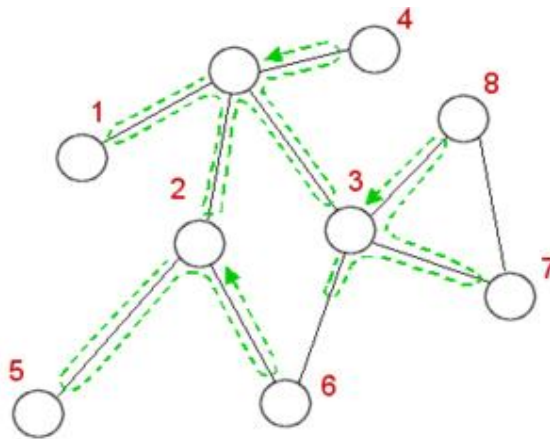
- Iš pradinės viršūnės turi būti pasiekiamos visos galinės viršūnės.
- Negali būti kabančių (neaktyvių) viršūnių.
- Negali būti amžinų ciklų (visuomet turi būti galimybė pereiti iš bet kurios viršūnės į kitą ir į galinę viršūnę).

Pagrindiniai grafo analizės algoritmai naudoja „paieškos gilyn“ arba „paieškos platyn“ metodus. Paieška į gylį (angl. *depth-first search DFS*) - tai paieškos grafe arba grafo apėjimo būdas, pasirenkama pradinė viršūnė ir einama grafo lankais kiek įmanoma gilyn, taip renkantis vis naują viršūnę. Kai pasiekama paskutinė aplankyta viršūnė ir neaplankytų kaimynių nebeturi, tuomet grįžtama yra iki artimiausio neaplankyto lanko, nuo kur vėl pradedama ieškoti gilyn. Paieška vykdoma kol yra surandama ieškomas tikslas arba kai bus aplankytos visos grafo viršūnės ir lankai (10 pav.) [17].



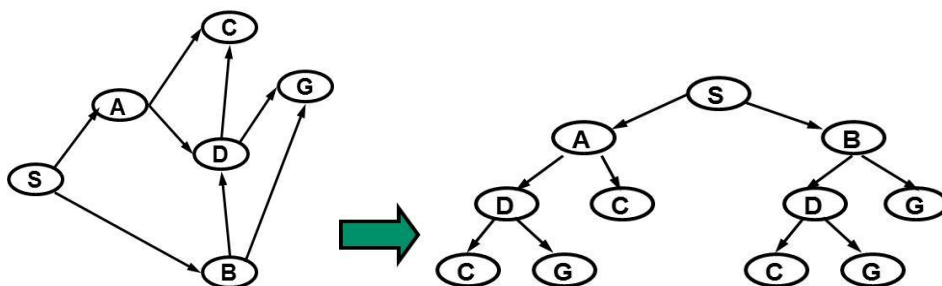
10 pav. Paieška į gylį [17]

Paieška į plotį (angl. *breadth-first search, BFS*) – tai paieškos grafe arba grafo apėjimo būdas, pasirenkama pradinė viršūnė, pirmiausia aplankoma visos jos kaimynės (tai viršūnės kurios yra sujungtos grafo lankais su pasirinkta viršūne). Taip analizuojamas visas pasirinktas grafai, kol randamas ieškomas tikslas arba kol yra apėinamos visos grafo viršūnės ir lankai (11 pav.) [18].



11 pav. Paieška į plotį [18]

Analizuojant grafo teisingumą panaudojant paiešką į plotį ar paiešką į gylį rezultatai atvaizduojami į medį (12 pav.).



12 pav. Grafo atvaizdavimas į medį [19]

Išanalizavus grafą, gaunamas rezultatas (medis), kuris parodo visus galimus kelius nuo pradinės iki galinės viršūnės. Grafo korektiškumas tikrinamas pagal aukščiau apibrėžtus kriterijus.

1.4. Pirmojo skyriaus išvados

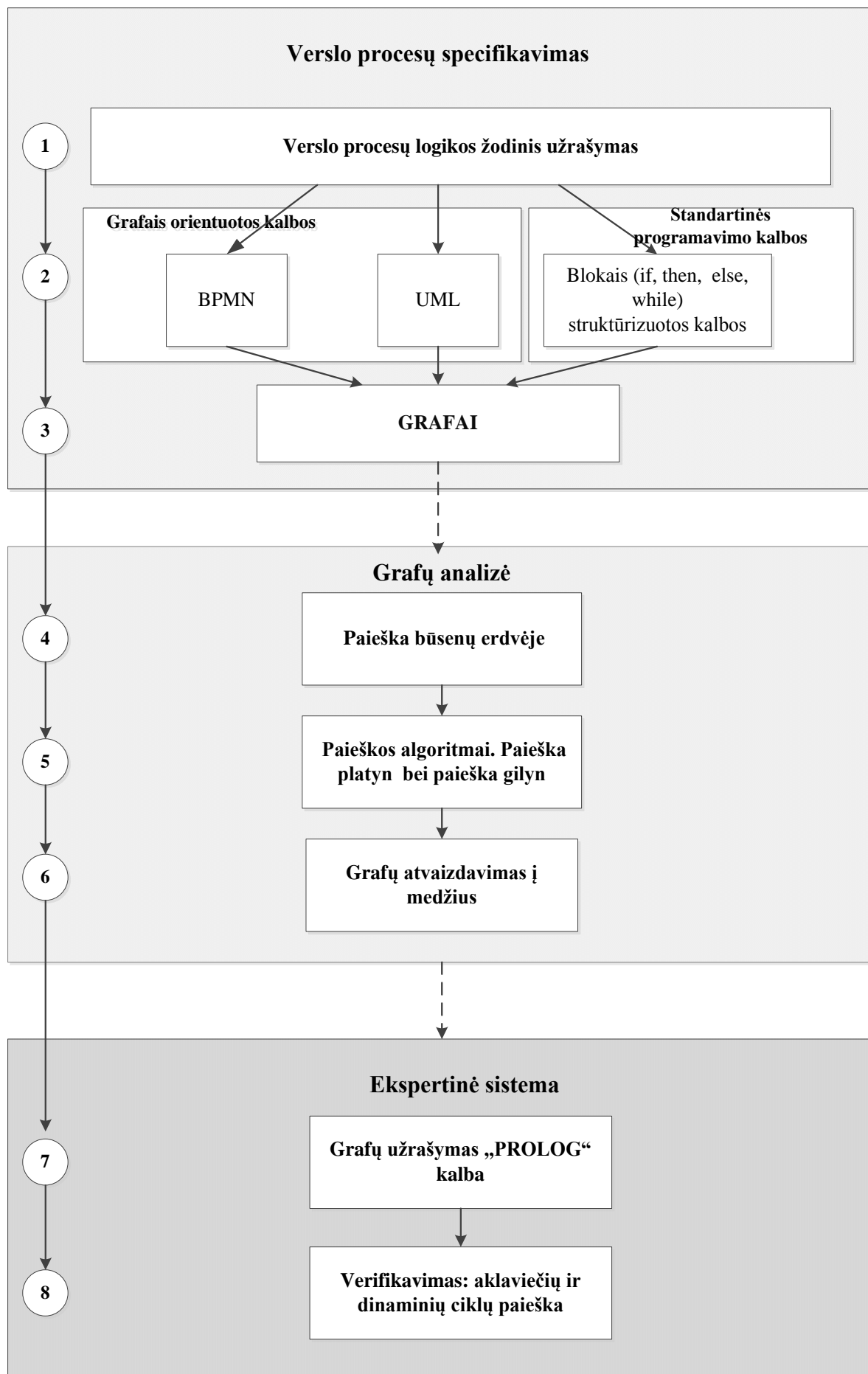
1. Atlikta mokslinių šaltinių analizė parodė, kad šiuo metu verslo procesų logikai aprašyti naudojami metodai (tokie kaip verslo procesų modeliavimo notacija (BPMN), unifikuota modeliavimo kalba (UML veiklos diagramos) bei kiti formalizmai yra gana sudėtingi ir, verslo atstovams, kurie nežino minėtų formalių procesų užrašymo būdų, sunku susišnekėti su informacinių technologijų specialistais modeliuojančiais šiuos procesus.

2. Be to, BPMN, UML veiklos diagramos ir kiti formalizmai naudoja sudėtingas programines sistemas, kad būtų galima patikrinti verslo procesų teisingumą.

3. Šiame darbe siūloma verslo procesų modeliavimui panaudoti grafus.

1.5. Siūloma magistrinio darbo problemos tyrimo metodika

Kompiuterizuojant verslo procesus bei kuriant darbų srautų valdymo sistemas pirmiausia reikia ① neformaliai aprašyti verslo procesų logiką. Antrame etape ② naudojami formalūs verslo procesų modeliai. Šiuo metu formaliai verslo procesų logikai aprašyti plačiausiai naudojami tokie metodai: UML veiklos diagramos, BPMN procesų diagramos arba tiesiog naudojamos struktūrizuotos kalbos (if, then else, while) ir kitos notacijos. Minėtų notacijų teisingumo tikrinimui reikalingos sudėtingos programinės priemonės. Darbe siūloma verslo procesus vaizduoti grafais, arba minėtas notacijas taip pat atvaizduoti grafais ③. Jų privalumas toks, kad jos leidžia grafiškai aprašyti sudėtingą procesų logiką. Grafų forma gerai suprantama tiek verslo tiek ir informacinių technologijų specialistams. Grafų analizei siūloma panaudoti grafų paieškos algoritmus ④ Paieškos sprendimas atvaizduojamas medžiais ⑤ ir ⑥. Visos analizės atlikimui siūloma panaudoti dirbtinio intelekto priemones – sukurti ekspertinę sistemą, kurioje grafų analizės duomenys užrašomi „Prolog“ kalba ⑦. Prolog kalboje yra realizuoti paieškos algoritmai leidžia aptikti verslo procesų aklavietes ir ciklus ⑧.



13 pav. Verslo procesų verifikavimo metodika

2. VERSLO PROCESŲ VERIFIKAVIMAS

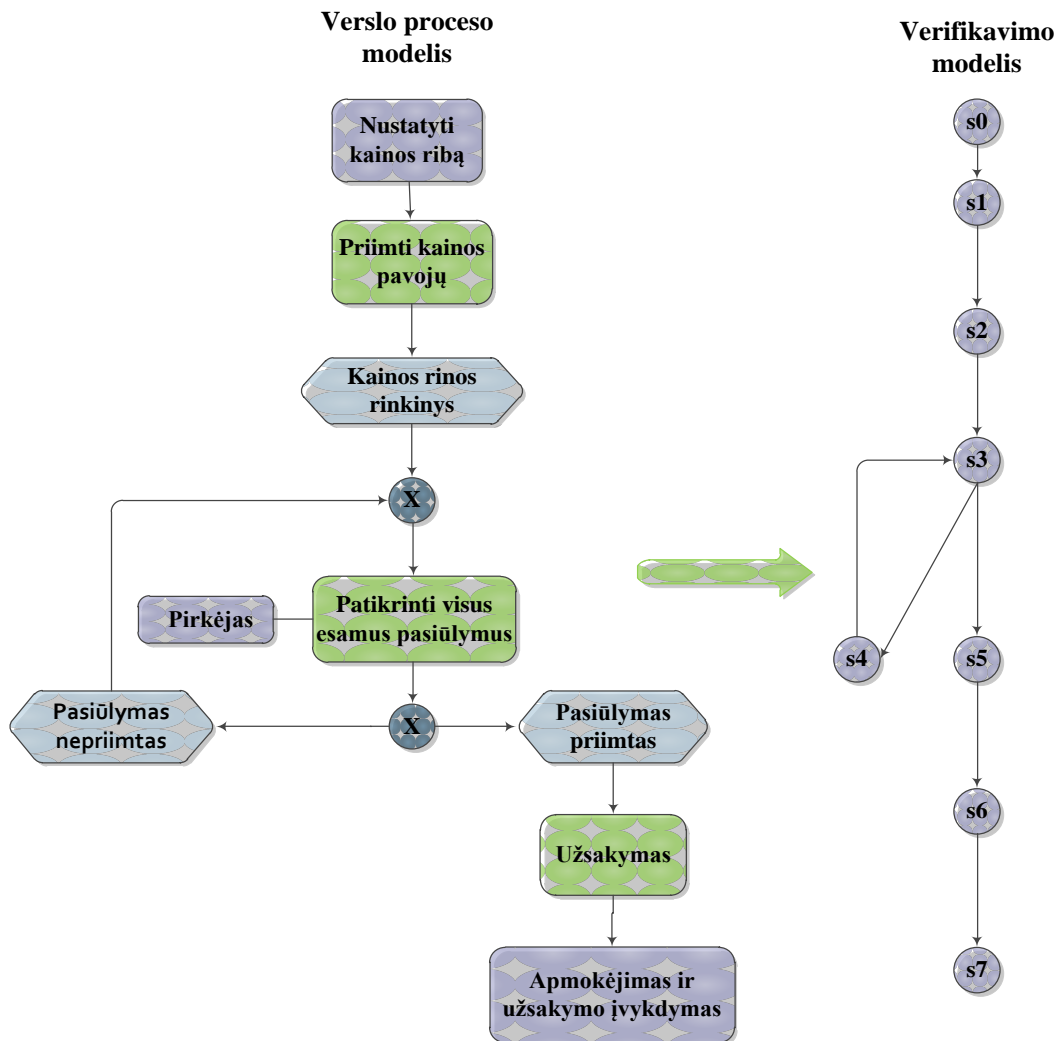
2.1. Pasiekiamumo savybių tyrimas

Projektuojant verslo sistemas viena iš pagrindinių projekto etapų yra verslo procesų verifikavimas. Apytikriai yra du formalus verifikavimo metodai [20].

Pirmasis būdas - yra modelio patikrinimas, kuris sudarytas iš sistemiskai nuodugnaus matematinio modelio patikrinimo. Įprastai tai sudaro visų būsenų ir perėjimų patikrinimas modelyje, naudojant tam tikras technikas apžiūrėti visas būsenų grupes. Savybės, kurias reikia patikrinti, dažnai yra užrašomos laiko logikos notacijomis. Antras metodas - yra loginis įrodymas (interferencija). Tai sudaro matematinio pagrindimo (įrodymo) apie sistemą formalios versijos, įprastai naudojant teoremą įrodančią programinę įrangą (pvz., tokią kaip HOL (angl., *Higher Order Logic*)). Įprastai tai yra iš dalies automatizuota ir naudotojo valdoma sistema [20], [21].

Šiandien verslo procesų verifikavimas dažnai atliekamas nuodugnaus matematinio modelio patikrinimo būdu. Kai kuriais atvejais verslo procesų įrankiai turi įmontuotą simulatorių, kuris paprastai naudojamas veiklos analizei (14 pav.) [16].

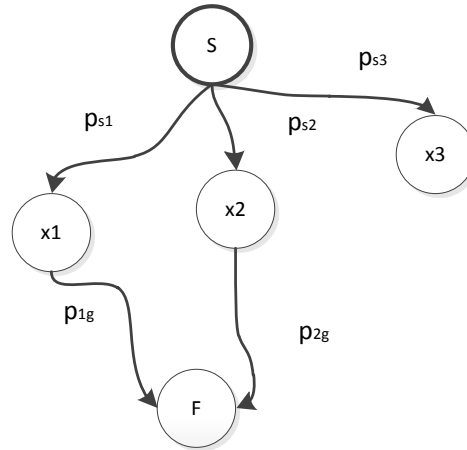
Šiame magistriniame darbe verifikavimui bus panaudoti abu aukščiau minėti metodai: visų būsenų ir perėjimų patikrinimas modelyje bus atliekamas panaudojant teoremą įrodančią programinę įrangą – Prolog- loginio išvedimo kalbą.



14 pav. Verslo diagramų verifikavimas [16]

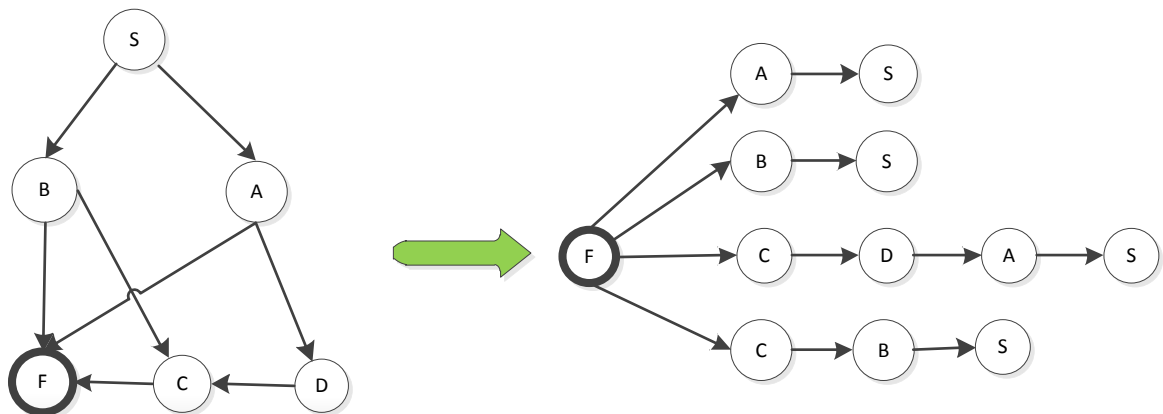
Nors verifikavimui yra taikoma daug metodų bei technikų, šiame darbe daugiausiai bus aptarinėjama verifikavimo savybių, kurias turi tenkinti sistemos būsenos. Pasiekiamumo analizė t.y., aklaviečių bei uždarų ciklų nebuvimas yra tipiškos savybės, kurias yra būtina verifikuoti.

Šią problemą spręsimė pasinaudojant pasiekiamumo paieškos algoritmus grafuose. Pateiktame grafe paveiksle (15 pav.) pavaizduota problema: reikia išanalizuoti visus kelius kurie prasideda pradžios taške S ir baigiasi pasirenkame tikslo taške F. Visi esantys keliai - p_j nuo pradžios viršūnės S iki tikslo mazgo F gali būti išskaidomi į $S \xrightarrow{p_{sj}} X_j \xrightarrow{p_{jg}} F$, $j=1,2,3$. Pavyzdžiui iš x_3 viršūnės neįmanoma pasiekti tikslo viršūnės F, jei pradėsime iš starto būsenos S. O tai reiškia, kad įvyksta pasiekiamumo ypatybių pažeidimas - grafas netenkina pasiekiamumo savybės sąlygų [9].



15 pav. Pasiekiamumo iliustravimas

Tokias problemas galima bandyti spręsti pasinaudojus kelio radimo algoritmus grafe (buvo aprašyti pirmame skyriuje). Kelio radimo algoritmai, randa trumpiausią kelia nuo pradžios viršūnės iki tikslo viršūnės. Siekiant, kad būtų surasti visi keliai algoritmas yra modifikuojamas. Modifikuotame algoritme taikoma paieška gilyn ir kiekvienai viršūnei randamas trumpiausias kelias iš „viršaus į apačią“ (angl. *Top down*) stiliumi. Iš „viršaus į apačią“ stiliumi grafas analizuojamas pradedant nuo tikslo viršūnės. Paieškos algoritmas randa visus pasiekiamus kelius iš tikslo mazgo F į pradžios mazgą S. Šis paieškos algoritmas gali analizuoti grafus kuriuose yra pasiekiamumo problemų. Šio algoritmo verifikavimo principas pavaizduotas 16 paveiksle [9].



16 pav. Modifikuotas paieškos algoritmas

Pasiekiamumo verifikavimas parodo ar verslo procesas turi tam tikrą pageidaujamą savybę ir tenkina visas numatytas sąlygas [20]. Dažniausiai tikrinama ar sistema nepatenka į aklavietes ar uždaruosius ciklus. Kitus reikalavimus galima užrašyti invariantais.

Aklavietė – tai sistemos būseną, į kuria galime patekti, tačiau iš jos niekada nepereinama į jokią kitą sistemos būseną. Būsenų grafe tai būseną, turinti įeinančią briauną tačiau neturinti išeinančių briaunų [9].

Uždaras ciklas – tai sistemos būsenų aibė, į kurią patekus neegzistuoja perėjimas į jokiais kitas esamas sistemos būsenas. Kitaip sakant, jei sistema patenka į šias būsenas, ji patenka į amžiną ciklą. Būsenų grafe tai stipriai susijusi komponentė, turinti daugiau nei vieną viršūnę ir neturinti išeinančių briaunų į kitas stipriai susijusias komponentes [9].

Invariantas – tai išraiška, aprašyta sistemos būsenos vektoriaus kintamaisiais, matematiniais veiksmiais, loginėmis išraiškėmis. Invariantas turi būti teisingas visoms sistemos būsenoms, nepriklausomai nuo įvykių [22].

2.2. Formalus verifikavimo uždavinio užrašymas

Apibrėžimas [9] [16]: Verslo procesas yra grafas $G=(X,W)$:

- $X = \{x_1, x_2, \dots, x_n\}$ yra viršūnių rinkinys, kuris apibrėžia verslo procesų užduotis. Viršūnė gali būti IR/ARBA tipo;
- $W \subseteq X \times X$ - lankų aibė, kuri vaizduoja užduočių srautus;
- Rinkinys $X^s \subseteq X$ yra baigtinis pradžios užduočių rinkinys;
- Kelias yra lankų seka $p = x_1, x_2, \dots, x_{k-1}, x_k$ ir visi lankai yra skirtingi $i = 1, 2, \dots, k-1, k$. Kelias yra ciklinis jei $x_i = x_j$;

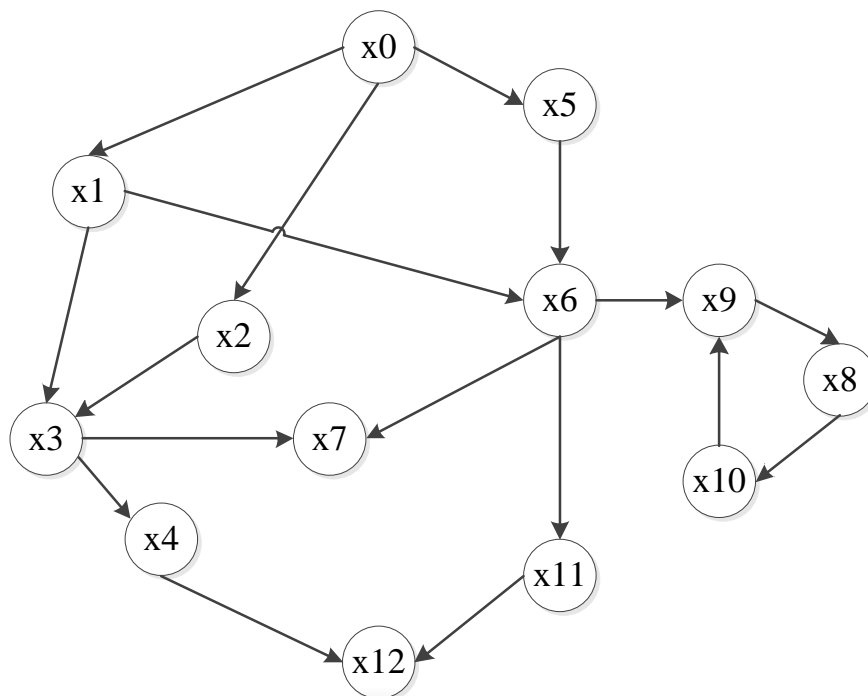
Grafą charakterizuoja:

- Viršūnės $x \in X$ aplinka yra gretimų viršūnių rinkinys žymimas $E(x) = \{u \in X : \{x, u\} \in W\}$. Du mazgai $u, v \in X$ yra gretimi jei jie sujungti kraštine. Viršūnių įėjimai $E^+(x)$ ir išėjimai $E^-(x)$ yra žymimi be $E(x) = E^+(x) \cup E^-(x)$;
- Viršūnės laipsnis yra aprašomas kaip $\deg(x) = |E(x)|$. Įėjimų laipsnis yra žymimas $\deg^-(x)$ ir išėjimų laipsnis žymimas $\deg^+(x)$. Laipsnis aprašo $\deg(x) = \deg^-(x) + \deg^+(x)$ įeinančias ir išeinančias kraštines iš ir į viršūnes;
- Viršūnė $x \in X$ su $\deg^-(x) = 0, \deg^+(x) > 0$ yra vadinamas pradžios arba pradinė viršūnė;
- Panašiai, tikslo viršūnė yra pažymėta kaip $\deg^+(x) = 0, \deg^-(x) > 0$;
- Kitos viršūnės $x \in X$ su laipsniais $\deg(x) = \deg^-(x) + \deg^+(x)$ yra vadinami vidinėmis grafo viršūnėmis.

Pasiekiamumo klaidos, kurios gali būti aptinkamos grafe, yra tokios:

- Aklavietė - yra vidinė viršūnė $x \in X$ kurios $\deg^+(x) = 0, \deg^-(x) > 0$ ir nepriklauso tikslo viršūnių rinkiniui;
- Nesibaigiantis ciklas - tai ciklinė viršūnių seka, dėl kurios atsiranda nepasiekiamų viršūnių.

Verslo proceso modeliavimui ir verifikavimui kaip pavyzdį aprašysime grafą (17 pav. Grafo pavyzdys [9]).



17 pav. Grafo pavyzdys [9]

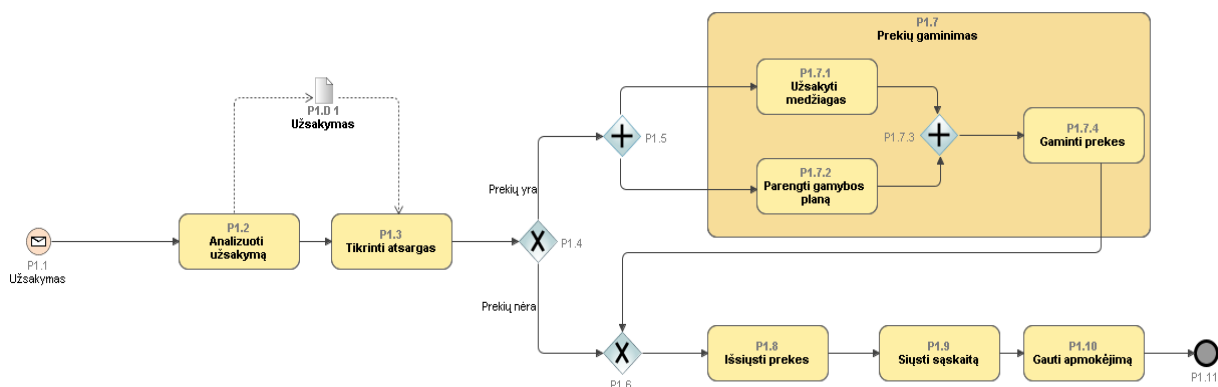
- Duoto grafo viršūnių rinkinys yra $X = \{x_0, x_1, x_2, \dots, x_{12}\}$;
- Grafo lankų rinkinys $W = \{(x_0, x_1), (x_0, x_2), (x_0, x_5), (x_1, x_3), (x_1, x_6), (x_2, x_3), (x_3, x_4), (x_3, x_7), (x_4, x_{12}), (x_5, x_6), (x_6, x_7), (x_6, x_9), (x_6, x_{11}), (x_7, x_{12}), (x_8, x_{10}), (x_9, x_8), (x_{10}, x_9), (x_{11}, x_{12})\}$;
- Pradžios viršūnė yra $\{x_0\} \subset X$ kur $\deg^-(x_0) = 0, \deg^+(x_0) > 0$;
- Tikslų viršūnė $\{x_{12}\} \subset X$ kur $\deg^+(x_i) = 0, \deg^-(x_i) > 0, i = 12$;

Klaidos aptiktos pateiktame grafe:

- Aklavietė yra tarpinėje viršūnėje $\{x_7\} \subset X$, kurios $\deg^+(x_7) = 0, \deg^-(x_7) > 0$. Ši viršūnė nepriklauso tikslo viršūnių aibei.
- Nesibaigiančiam ciklui priklauso šios viršūnės $\{x_8, x_9, x_{10}\} \subset X$.

2.3. Verslo proceso „Užsakymas gamybai“ verifikavimo pavyzdys

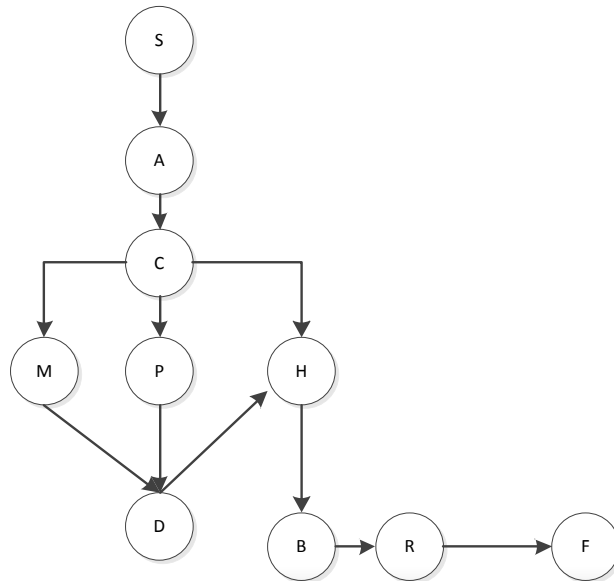
Paveiksle (18 pav.) pateikiamas verslo proceso pavyzdys [16] naudojantis BPMN notaciją, kuriame vaizduojamas užsakymo procesas.



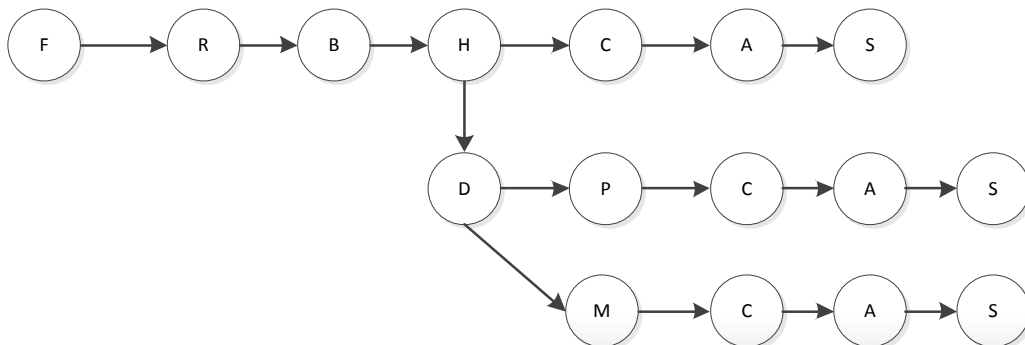
18 pav. Verslo proceso pavyzdys naudojantis BPMN notaciją [13]

Pavyzdyje matomi pagrindiniai notacijos elementai: įvykis, veiklos, srautas. Proceso modelis prasideda nuo veiklos pradžios įvykio. Seka veiklų skirtų išanalizuoti užsakymą ir patikrinti prekių atsargas yra įvykdomi prieš sprendimo tašką, kuris žymimas simboliu "X". Jei užsakomų prekių yra sandėlyje, tada einama apatine šaka. Kitu atveju prekės pirmiausia turi būti pagamintos, todėl viršutinė šaka yra pasirenkama.

Ši BPMN notacija gali būti transformuota į grafą, todėl šios BPMN diagramos procesą galima atvaizduoti tokiu grafu (19 pav.). Paveiksle atitinka: S – pradžia, A – analizuoti užsakymą, C – patikrinti atsargas, P – pirkti medžiagas, M – sudaryti gamybos planą, D – pagaminti prekes, H – išsiųsti prekes, B – siųsti sąskaitą, R – gauti apmokėjimą, F – baigti. Paveiksle (20 pav.) pateikta visų galimų kelių paieška.



19 pav. BPMN diagramos procesas atvaizduotas grafu



20 pav. Paieškos procesas atvaizduotas medžiu

2.4. Antrojo skyriaus išvados

1. Šiame darbe pasiūlyta verslo procesų modeliavimui panaudoti grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose.

2. Šio metodo privalumas toks, kad jis leidžia nesudėtingai aprašyti procesų logiką grafais, o tokia grafų forma gerai suprantama tiek verslo atstovams, tiek ir informacinių technologijų specialistams.

3. VERIFIKAVIMAS DIRBTINIO INTELEKTO SISTEMOSE

3.1. Dirbtinio intelekto taikymo sritys

Dirbtinis intelektas (toliau-DI) (angl. *Artificial Intelligence*) yra labai plati sąvoka, todėl yra apibrėžiamos kelios pagrindinės šios mokslo šakos sritys [23]:

- Neuroniniai tinklai – tai sistemos kurios imituoja žmogaus smegenų architektūrą.
- Natūralios kalbos sistemos – viena plačiai išvystytų dirbtinio intelekto tyrinėjimo sričių. Šiomis sistemomis siekiama, kad sugebėtų suprasti, interpretuoti ir bendrauti natūralia žmonių kalba. Natūralios kalbos sistemų formos taikomos tekstiniuose redaktoriuose, kompiuteriniuose žaidimuose, psichologinės pagalbos priemonėse bei daugelyje kitų sričių. Neretai natūralios kalbos sistemos pritaikomos pojūčių modeliavimo programose.
- Kompiuteriniai žaidimai – šioje srityje dirbtinio intelekto tikslas sukurti kompiuterinius žaidimus, kurie būtų žmogiškojo oponento ekvivalentas ir turėtų kuo labiau natūralizuotą virtualios aplinkos sąveiką su vartotoju.
- Pojūčių modeliavimas – dirbtinio intelekto priemone yra bandoma imituoti žmogaus suvokimą į aplinką arba dirgiklių interpretacijos procesus. Kuriamos programos, kurios atpažįsta sudėtingus garsus, vaizdus, kuriuos taip pat gali apdoroti ir panaudoti. Galima teikti, kad tai yra labiausiai komerciniu požiūriu išplėtotą dirbtinio intelekto tyrinėjimo sritis ir daugiausiai pritaikomos grafiniuose redaktoriuose, fotografijos įrenginiuose, apsaugos sistemose, seismologijoje, astronomijoje.
- Ekspertinės sistemos (ES) - yra dirbtinio intelekto sistemos, kurios kuriamos konkrečios probleminės srities uždavinių sprendimui remiantis šios srities ekspertų žiniomis.

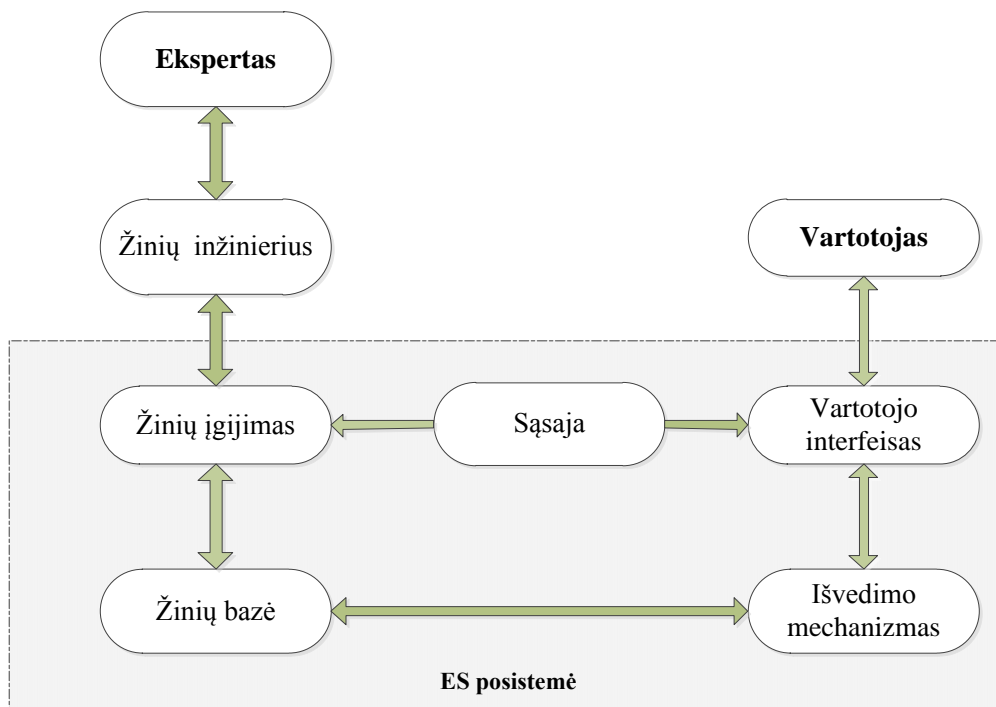
Šiame magistriniame darbe kuriama ekspertinė sistema, skirta ankstesniuose skyriuose aprašytų problemų sprendimui.

Dirbtinio intelekto sistema, sukurta konkrečios probleminės srities uždavinių sprendimui, vadinama ekspertine sistema. Apibrėžiama, kad ekspertinė sistema (ES) tai yra įvairios sprendimų paramos sistemos, kurios naudoja žinias, procedūrų išvadas sprendžiant sudėtingas iškilusias problemas. Kuriamos ES geba priimti sprendimą ir paaiškinti pasirinkto sprendimo priežastis. ES skirstomos į dvi rūšis: sisteminės ir pagrįstos taisyklėmis. Plačiausiai yra naudojama taisyklėmis pagrįstos sistemos, todėl apie šią rūšį bus kalbama plačiau. Dažniausiai kuriamas ES sudaro penki pagrindiniai elementai tai [24]:

- žinių bazė, ar ekspertų žinios;
- išvedimo mechanizmas (tai pritaikytas algoritmas, kuris suteikia galimybę modeliuoti žmogaus sprendžiamas problemas procesą);
- vartotojo sąsaja, kuri leidžia apžvelgti duomenis apie sprendžiamas problemas elementus;
- paaiškinamasis posistemis (suteikia informaciją apie sprendžiamą klausimą);
- žinių rinkimo posistemis (tikslas tirti eksperto išsilavinimo lygį, papildant jo turimas žinias ir taip pat parenkantis priemones ekspertinei sistemai kurti). Ekspertinės sistemos veiklos lygis priklauso nuo žinių bazės funkcijų dydžio ir informacijos kokybės.

Ekspertinės sistemos, žiniomis pagrįstos ekspertinės sistemos, panaudoja ekspertų žinias todėl, kad būtų sprendžiamos problemos, kurios paprastai reikalauja žmogaus intelekto pagalbos. Todėl ekspertinės sistemos naudoja ekspertizės žinias kaip duomenis arba kaip taisykles kompiuterio viduje. Taisyklės arba reikalingi duomenys iškviečiami tuomet, kai reikia spręsti atitinkamas problemas [25], [26].

Taisyklių vaizdavimui dažnai yra panaudojamos duomenų bazės bei programavimo kalbos, į kurias šios žinios įtraukiamos kaip programos kodo dalis. Pasikeitus žinioms, programa taip pat turi būti pakeista bei atnaujinta. Kai yra naudojamos tokios programavimo kalbos kaip Prolog, palaikančios priemones taisyklių užrašymui, tai žiniomis pagrįstos sistemos remiasi nedideliais informacijos fragmentais, kuriant žinių bazes, o vėliau pastarosios žinios tarnauja atitinkamų problemų sprendimui. Paprasčiausios ekspertinės sistemos, sudarytos taisyklių pagrindu, struktūra parodyta (21 pav.) [25], [27].



21 pav. Taisyklėmis pagrįsta ES struktūra [27]

Taisyklėmis grįstos ES pasižymi išskirtiniais bruožais tokiais kaip, kad jos sudarytos iš išvedimo mechanizmo ir žinių bazės. ES žinių bazė apjungia taisykles, kurios nusako žinias apie tam tikrą probleminę sritį. Išvedimo mechanizmas pateikia, galimybę padaryti loginę išvadą iš turimų žinių bazės duomenų. Dažniausiai išvedimo mechanizmas yra nepriklausomas nuo žinių bazės, ir tai įgalina nesunkiai pakeisti pastarąją nauja žinių baze, nedarant pakeitimų išvedimo mechanizme [22], [27]. Žinių bazė ne visais atvejais apibrėžta – ji gali apjungti taisykles, kuriose nurodomos tikimybės.

Daugeliu atveju nemažas dėmesys yra skiriamas vartotojo sąsajai ir sistemos dialogui su vartotoju. Jos turi galimybę ne tik daryti išvadas iš turimų žinių bazės duomenų, bet ir paaiškinti kaip ir kodėl atitinkami sprendimai buvo priimti [27], [28].

Nesunku pastebėti, kad pirmoji iš paminėtų savybių sutampa su loginio programavimo koncepcija, t.y. atskiria loginę komponentę (kokia yra problemos esmė) nuo valdymo komponentės (kaip problema turėtų būti išspręsta). Tai galėtų būti išreikšta tokios formulės pagalba *Ekspertinė Sistema = Žinių Bazė + Valdymas + Vartotojo Interfeisas* [28].

3.2. Ekspertinių žinių vaizdavimas produkcijų sistema

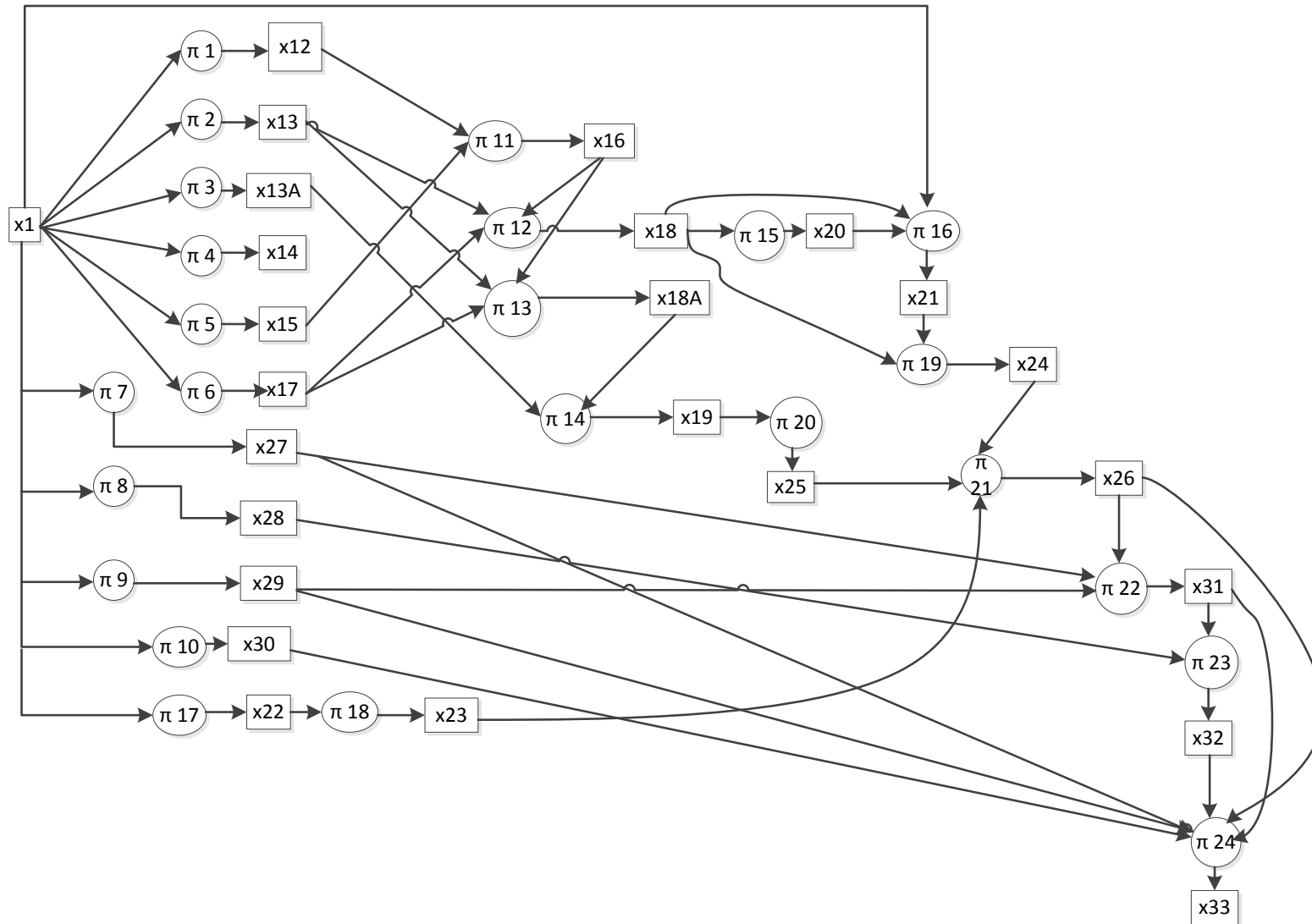
Kuriant ekspertines sistemas vienas iš projektavimo etapų yra - eksperto žinių įvedimas į sistemą. Šios žinios dažnai būna platūs, nevienareikšmiai, ne griežtai apibrėžti samprotavimai, padiktuoti toje srityje dirbančių ekspertų. Siekiant efektyvumo šios žinios turi būti transformuojamos į matematinį jų atvaizdavimą. Vienas žinių vaizdavimo modelių galėtų būti – produkcijų sistema [28], [29].

Produkcijų uždaviniai formuojami taip – randama produkcijos seka, kuri pradinę būseną perveda į terminalinę būseną. Terminalinė būseną – tai būseną kurioje yra tikslo objektas. Kad gautume produkcijų sekas yra pritaikomi du algoritmai tai tiesioginis išvedimas (angl. *forward chaining*) kuris pradeda nuo pradinių duomenų ir juda link galutinės, ir atbulinis išvedimas (angl. *backward chaining*) kuris pradeda nuo galutinės ir juda link pradinės [29].

Kaip pavyzdį („Gyventojų pajamų/ mokesčių deklaracijos“ grafa, kuris bus pateiktas tolesniuose skyriuose) panagrinėsime produkcijų sistemą:

$\pi_1:$ $x1 \rightarrow x12$
 $\pi_2:$ $x1 \rightarrow x13$
 $\pi_3:$ $x1 \rightarrow x13A$
 $\pi_4:$ $x1 \rightarrow x14$
 $\pi_5:$ $x1 \rightarrow x15$
 $\pi_6:$ $x1 \rightarrow x17$
 $\pi_7:$ $x1 \rightarrow x27$
 $\pi_8:$ $x1 \rightarrow x28$
 $\pi_9:$ $x1 \rightarrow x29$
 $\pi_{10}:$ $x1 \rightarrow x30$
 $\pi_{11}:$ $x12, x15 \rightarrow x16$
 $\pi_{12}:$ $x13, x16, x17 \rightarrow x18$
 $\pi_{13}:$ $x13, x16, x17 \rightarrow x18A$
 $\pi_{14}:$ $x13A, x18A \rightarrow x19$
 $\pi_{15}:$ $x18 \rightarrow x20$
 $\pi_{16}:$ $x20, x18, x1 \rightarrow x21$
 $\pi_{17}:$ $x1 \rightarrow x22$
 $\pi_{18}:$ $x22 \rightarrow x23$
 $\pi_{19}:$ $x18, x21 \rightarrow x24$
 $\pi_{20}:$ $x19 \rightarrow x25$
 $\pi_{21}:$ $x23, x24, x25 \rightarrow x26$
 $\pi_{22}:$ $x27, x26, x29 \rightarrow x31$
 $\pi_{23}:$ $x28, x31 \rightarrow x32$
 $\pi_{24}:$ $x30, x29, x27, x32, x31, x26 \rightarrow x33$

Atliekant išvedimą ši produkcijų seka gali būti transformuota į *dvidalį grafa* (angl. *bipartite graph*), pateiktą (22 pav.). Tokiame grafe viršūnės yra dviejų rūšių: tai viršūnės, vaizduojamos stačiakampiais, ir viršūnės, vaizduojamos skrituliais. Viršūnė-stačiakampis vaizduoja objektą ir yra pažymėta šio objekto vardu. Viršūnė-apskritimas vaizduoja produkciją ir yra pažymėta šios produkcijos vardu. Konceptiniame modeliavime tokio tipo grafas yra vadinamas *semantiniu grafu* (semantiniu tinklu, konceptiniu grafu). Jis vaizduoja ir objektus, ir produkcijas. Išvedimas produkcijų sistemoje pavaizduotas 22 paveiksle [29].



22 pav. Išvedimas produkcijų sistemoje

3.3. Ekspertinių žinių valdymas loginio programavimo kalba PROLOG

Viena iš dažniausiai ir labiausiai ekspertinių sistemų (ES) kūrimui naudojamų neprocedūrinių kalbų yra Prolog programavimo kalba, kuri glaudžiai siejasi su teiginių logika. Kalba kuri gali nesunkiai aprašyti realaus gyvenimo situacijas programiškai. Pavadinimas - Prolog yra kilęs iš anglų “*PRO*gramming in *LOG*ic”, kuris reiškia - programavimas logikoje [27].

Prolog programavimo kalba nėra panaši į kitas programavimo kalbas, nes Prolog kalba dažniausiai yra nenuosekli veiksmų seka, bei faktai ir taisyklės, kurie leidžia padaryti reikiamas išvadas, visuma. Todėl ji priskiriama prie deklaratyviųjų kalbų. Prolog kalboje tam tikros išvados gaunamos iš jau žinomų faktų, kurie yra saugomi žinių bazėje. Ši programavimo kalba paremta Horno teiginiais – tai yra formalios sistemos, kitaip vadinami predikatų logika, poaibis kur vartojama supaprastintą predikatų logikos sintaksę, priartinančią šią kalbą prie natūralios kalbos paprastumo [27], [30].

Prolog programavimo kalbos išvedimo mechanizmas veikia kaip loginis mąstymas, kuris remiasi turima informacija iš žinių bazės. Išvedimo mechanizmas iš žinių bazės žiniomis pasinaudoja pasiremdamas atitikimo šablono principu – sulygina atsakymą su klausimu. Tokiu metodu yra pasiekama, kad mechanizmas apklausdamas apibrėžtą faktų ir taisyklių aibę, bando įrodyti, jog hipotezė teisinga. Ši programavimo kalba pasižymi ne tik loginiu mąstymu, atliekant atsakymų paiešką, bet gali surasti ir kelias alternatyvas, kurios tenkina tą pačią užklausą. Prolog programavimo kalba skiriasi nuo kitų programavimo kalbų ne vien savo naudojimo mechanizmu, bet ir nevykdo tiesioginio programos vykdymo - nuo pradžios iki galo, o naudoja sugrįžimo (*angl. backtracking*) mechanizmą, kurio dėka galima grįžti į bet kurią programos vietą, siekiant surasti daugiau nei vieną problemos sprendimo variantą [27]. Prolog programavimo kalboje, skirtingai nei kitose programavimo kalbose, nėra priskyrimo sąvokos. Prolog kalba parašytose programose kintamieji įgauna reikšmes tik tuo atveju, jei jie atitinka faktuose ir taisyklėse pateiktas konstantas ir, kol kintamasis neturi reikšmės, sakoma, kad jis laisvas. Kai kintamasis įgauna tam tikrą reikšmę, sakoma, jog jis tapo susietas. Bet kuris programos kintamasis gali būti susiejamas su konkrečia reikšme laiko tarpui, kurio reikia užklaustos atsakymui gauti. Po to šis susiejimas panaikinamas, ir Prolog interpretatorius grįžta atgal (*backtracking*), siekdamas surasti alternatyvų problemos sprendimą. Taigi Prolog kalboje kintamieji vartojami ne kaip informacijos saugykla, o kaip viena atitikimo šablono mechanizmo sudėtinių dalių [27], [31].

Prolog kalba palaiko ir užklaustos mechanizmą, leidžiantį užduoti klausimus, į kuriuos atsakymai pateikiami remiantis turimais faktais. Užklausą dar galima vadinti tikslu, nes pateikiant sistemai užklausą, iš tiesų yra nurodomas tikslas, kuris turėtų būti pasiektas. Loginiame programavime tikslai skirstomi į paprastus bei sudėtinius, o atskira tikslo dalis vadinama potiksliu. Todėl galima teigti, kad sudėtinis tikslas susideda iš dviejų arba daugiau potikslių. Toks sudėtinis tikslas yra patenkinamas, jei randamas atitikimas tarp visų tikslo potikslių bei informacijos, saugomos žinių bazėje. Be to, Prolog kalba turi keletą privalumų [27]:

- Loginio programavimo kalboje Prolog parašytose programose yra atskiriamos algoritmo loginė ir valdymo dalys. Šis atskyrimas leidžia specifikacijas lengvai modifikuoti, keičiant formulių aibę, aprašančią loginę dalį, nekeičiant valdymo dalies, bei atvirkščiai.
- Loginio programavimo kalba Prolog užrašytos specifikacijos yra ne tik formalios, bet ir vykdomos dėl Prolog interpretatoriaus mechanizmo, kuris užtikrina automatinę teoremų įrodymo galimybę.

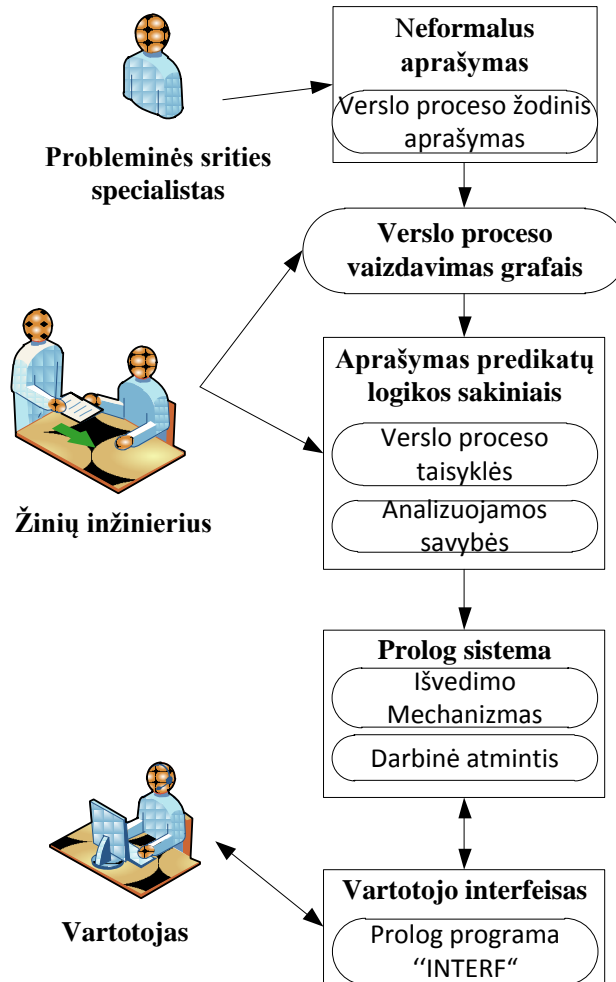
Šiame darbe sukurta ekspertinės sistema panaudojant loginio programavimo kalbą Prolog. Ši ekspertinė sistema analizuoja verslo procesus automatinio įrodymo rezoliucijų metodu, kuris yra loginio programavimo kalbos Prolog pagrindas.

3.4. Verslo procesų verifikavimo ekspertinė sistema

3.4.1. Verslo procesų verifikavimo ekspertinės sistemos dedamosios

Darbe pateikiamos verslo procesų verifikavimo metodikos realizavimui sukurta ekspertinė sistema. Šioje sistemoje verslo procesų verifikavimas vykdomas tokiais etapais:

1. Neformalus verslo proceso aprašymas.
2. Verslo proceso vaizdavimas grafais.
3. Aprašymas predikatų logikos sakiniais (PROLOG sakiniai).
4. Verslo procesų verifikavimas per vartotojo sąsają (INTERF).



23 pav. Verslo procesų verifikavimo sistemos struktūra

Ekspertinės sistemos modelyje, kaip pateikta (23 pav.) matyti ryšiai kurie siejasi su: probleminės srities specialistu, žinių inžinieriumi, potencialiu sistemos vartotoju [26, 27].

Pradiniam ekspertinės sistemos kūrimo etape svarbiausias vaidmuo tenka probleminės srities specialistui ar jų grupei (šių specialistų skaičius priklauso nuo sprendžiamos problemos masto). Probleminės srities specialistas – tai srities, kurioje specializuosis kuriama ekspertinė sistema, ekspertas, pateikiantis neformalų sprendžiamos problemos aprašymą [26].

Nors sprendžiamos problemos formulavimas, pateiktas natūralia kalba, yra labai svarbus, žinių inžinieriaus atliekamų funkcijų kokybė didžia dalimi lemia pačios ekspertinės sistemos realizavimo tikslumą bei efektyvumą. Žinių inžinierius privalo deklaratyvia forma koduoti probleminės srities specialistų pateiktą informaciją bei užtikrinti tikslų sistemos funkcionavimo ir atitinkamų specifikacijų atitikimą [26, 27].

Vartotojas – tai trečioji specialistų grandis, tiesiogiai susijusi su kuriama ekspertine sistema. Vartotojas – tai tam tikros specialybės atstovas, kuris savo darbe naudosis jau sukurta sistema, norėdamas gauti patarimą, tolygų probleminės srities eksperto išvada [26].

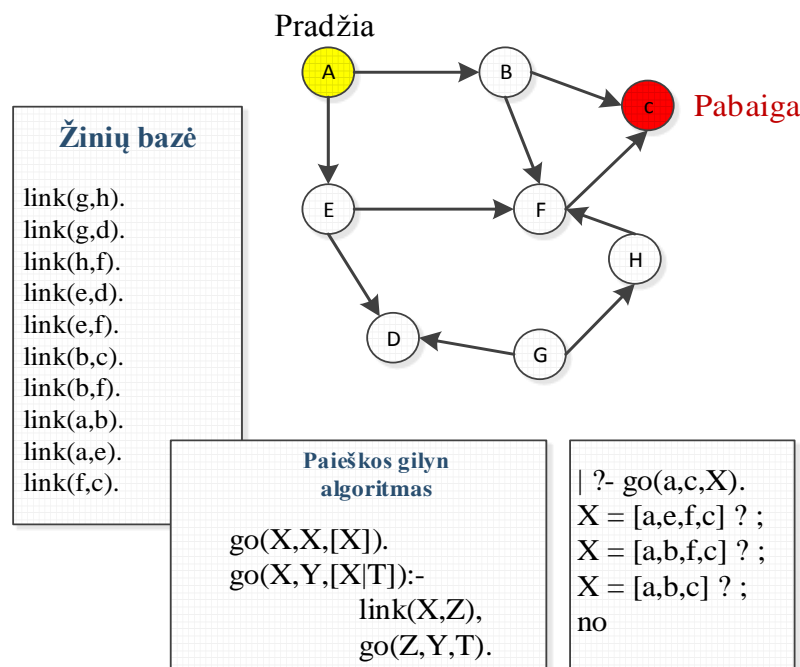
Apžvelgus kuriama ekspertine sistema ir jos veikėjus akivaizdu, kad probleminės srities specialistas bei vartotojas darbai nepriklauso vienas nuo kito. Tačiau žinių inžinierius kontroliuoja bei valdo ekspertinę sistemą projektavimo metu. Kadangi žinių inžinierius privalo koordinuoti probleminės srities specialisto darbą bei palaikyti ryšį su vartotoju, siekdamas patenkinti jo poreikius [26].

3.4.2. Paieška būsenų erdvėje realizuota Prolog kalba

Kaip jau buvo minėta ankstesniuose skyriuose, daugelio grafų teorijos uždavinių sprendimo algoritmų pagrindą sudaro sisteminga grafo viršūnių peržiūra, t.y. toks grafo viršūnių apėjimas, kad kiekviena viršūnė nagrinėjama vienintelį kartą.

Kelio paieškos problema aprašoma tokiais komponentais [32]:

- Pradinė būsena. Pradinė būsena nurodo paieškos pradžios viršūnę.
 - Galimų perėjimų aibė. Tai visi perėjimai į įpėdinio būsenas.
 - Būsenų erdvė. Tai visų galimų būsenų aibė. Pradinė būsena kartu su galimų perėjimų sąrašu apibrėžia kelio paieškos būsenų erdvę.
 - Kelias būsenų erdvėje apibrėžiamas kaip būsenų seka, sujungta galimų perėjimų seka.
 - Tikslo arba galinė būsena. Apibrėžia siekiamą tikslo būseną.
- Kaip šis uždavinys formuluojamas Prolog kalba yra pavaizduota 24 paveiksle.



24 pav. Paieškos uždavinio užrašymas Prolog kalba [32]

Prolog programą sudaro faktai ir taisyklės [32]:

1. Žinių bazėje surašomi visi galimi perėjimai (lankai) tarp būsenų (grafo viršūnių).

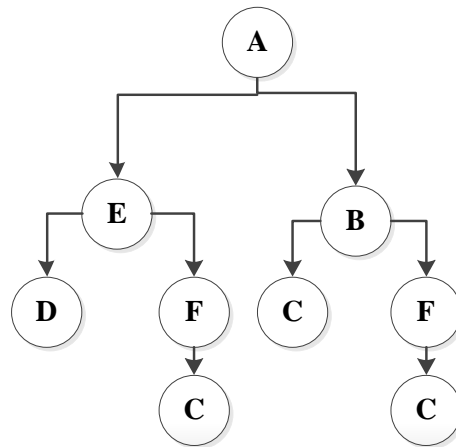
Predikatas $\text{link}(X,Y)$ aprašo perėjimą tarp X ir Y būsenų. Pvz., faktas:

- $\text{link}(a,e)$ išreiškia ryšį tarp dviejų būsenų a ir e.

2. Taisyklė, aprašanti būdą, kaip sugeneruoti visas įmanomas būsenas iš duotosios X būsenos į būseną Y per tarpines viršūnes Z.

- $go(X,Y,[X|T]) :- link(X,Z), go(Z,Y,T).$
3. Taisyklė nusakanti paieškos pabaigą.
- $go(X,X,[X]).$
4. Konsultavimas (klausimas): ar galima patekti iš būsenos **a** į būseną **c**. Kelias išsaugomas sąrašė **X**.
- $?- go(a,c,X)$

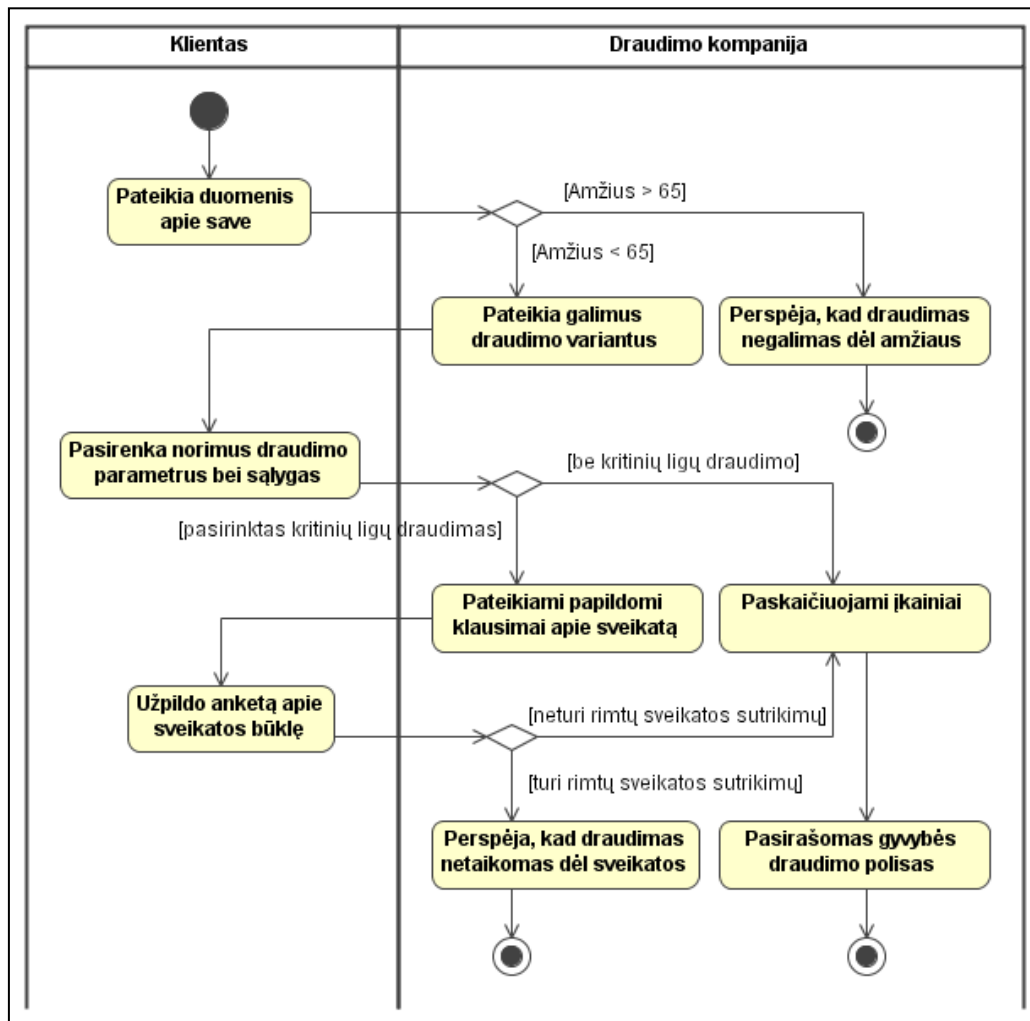
25 paveiksle pavaizduotas išvedimo procesas, kai naudojamas paieškos gilyn algoritmas.



25 pav. Paieškos gilyn grafinis vaizdavimas [32]

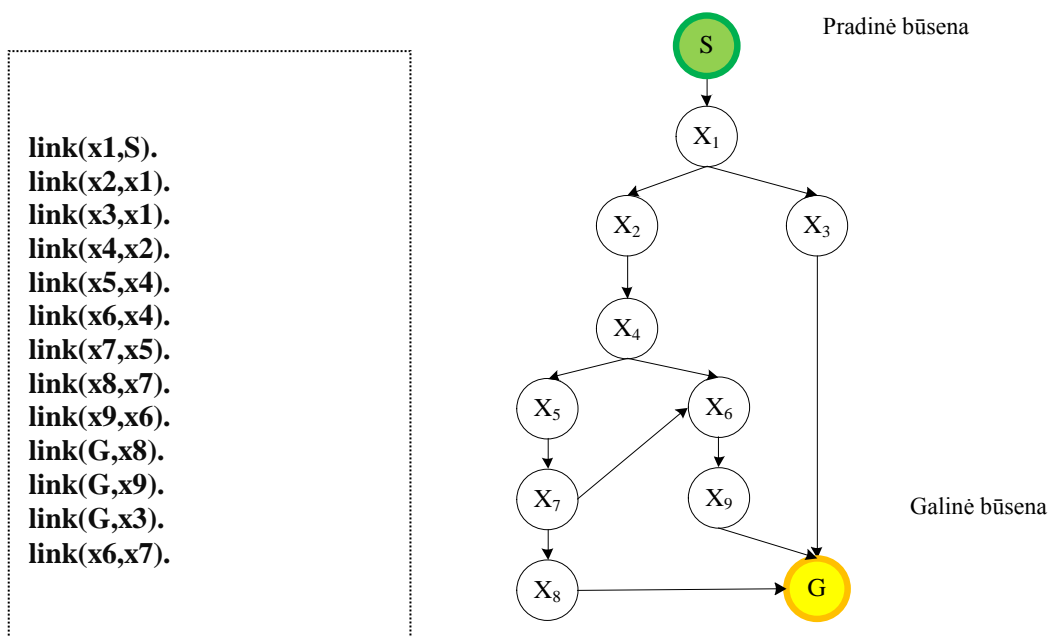
Apžvelgus paiešką būsenų erdvėje, kuri realizuojama su Prolog kalba, pritaikome pasirinktam realiam uždaviniui spręsti. 26- 28 paveiksluose pateiktas verslo proceso „Gyvybės draudimo procesas“ UML diagrama, grafas ir sprendimo medis.

UML veiklos diagramoje yra pavaizduota kaip vyksta draudimo procesas: kai žmogus pateikia informaciją apie save draudimo kompanijai, kur draudimo kompanijos atstovas peržiūri pateiktą informaciją ir pagal įmonės vidines taisykles nustato, ar pateiktas prašymas galimas ar ne (kadangi yra žmogaus metų limitas, kuris negali būti viršytas jei norima apsidrausti sveikata) [9].

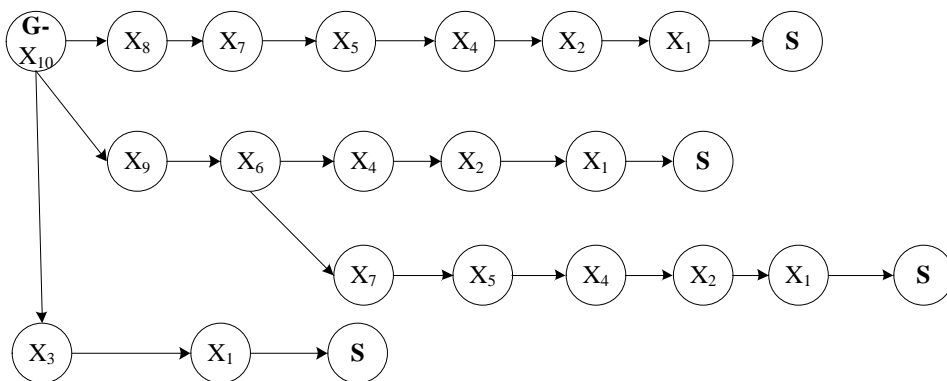


26 pav. Gyvybės draudimo UML veiklos diagrama [9]

Analizuodami toliau šį gyvybės draudimo procesą jį pavaizduojame į orientuotą grafą (27 pav.).



27 pav. Verslo proceso grafas



28 pav. Verslo proceso sprendimo medis

3.4.3. Prolog sakiniai, tikrinantys aklaviečių ir ciklų nebuvimą

Ciklų paieškai užrašomos taisyklės. Pirmoje taisyklėje sprendimo procesas pradedamas nuo pradinės s viršūnės iki tikslo viršūnės g.

```
execute:- write('Kelių be ciklų paieška:'),nl,nl,
          go(g,s,X),
          nl,
          write('X= '),
          write(X),
          nl,
          fail.
execute:- nl,
          go_2(g,s,X),
          nl,
          fail.
```

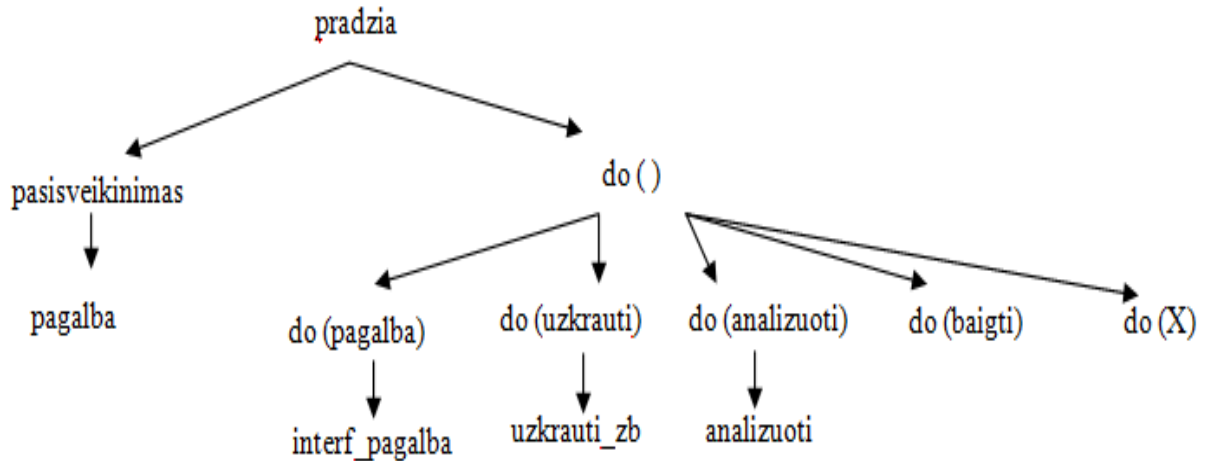
Antroji taisyklė surastas viršūnes, kurios patenka į aklavietes ar ciklus surašo į sąrašą:

```
execute:- nl,write('Viršūnės patekusios į ciklus ar
aklavietes: '),nl,
          findall(V1,link(V1,V2),LV1),
          write(LV1),nl,
          fail.
```

3.4.4. Ekspertinės sistemos vartotojo sąsaja

Vartotojo sąsaja - skirta bendravimui tarp vartotojo ir programos. Tokį dialogą tarp sistemos ir vartotojo sukuria "INTERF" programa. Ji naudojama žinių bazei pateikiant predikatus, paaiškinimus ar atsakymus į vartotojo klausimus.

Magistrinio darbo vartotojo sąsaja "INTERF" yra sukurtas su tokiais predikatais (29 pav.).



29 pav. Vartotojo sąsajos INTERF predikatai

Pagrindinis "INTERF" predikatas yra "pradzia". Kuris sukuria pradžią mūsų kuriamoje ES. Pagrindinį predikatą sudaro:

```

pradzia :-
  pasisveikinimas,
  repeat,
  write('> '),
  read(X),
  do(X),
  X == baigti.

```

Pasileidus kuriamą ES ir suvedus pirmą predikatą "pradzia", kuris į ekraną atspausdina pranešimą, kad prisijungėte prie vartotojo sąsajos. ES sistema pasisveikina su vartotoju, pateikia informaciją apie Prolog apvalkalą. Kad vartotojui būtų aiškiau ką reikėtų įvesti į ekraną yra pateikiama dar viena komanda "pagalba", kuri pateikia papildomos informacijos. Visi predikatai yra imami iš žinių bazės failų, kur yra nurodytos funkcijų komandos.

```

pasisveikinimas :-
  nl, write('-----'), nl,
  nl, write('Sveiki, Jūs naudojatės programa "Vartotojo sąsaja" !'), nl,
  nl, write('"Vartotojo sąsaja" - tai Prolog apvalkalas, '),
  nl,
  write('realizuojantis verslo procesų verifikavimą. '), nl,
  nl, write('-----'), nl,
  pagalba.
do(pagalba) :-
  interf_pagalba, !.
do(uzkrauti) :-
  uzkrauti_zb, !.
do(analizuoti) :-
  analizuoti, !.
do(kaip) :-
  kaip, !.
do(baigti)

```

Komanda "pagalba":

```
pagalba :-  
  nl, write('-----'), nl,  
  nl, write('Prašome užkrauti norimą žinių bazės failą. '), nl,  
  write('Tai atlieka komanda "uzkrauti". '), nl,  
  nl, write('Komanda "pagalba" Jums suteiks papildomą  
informaciją. '), nl,  
  nl, write('-----'), nl,  
nl..
```

Visi predikatai ar funkcijos yra rašomos į komandinės eilutes Prolog lange. Predikatai tokie kaip do() jau nurodo tikslią funkciją tarkim parašoma "pagalba." tai tokia užklausa - do(pagalba) :- interf_pagalba, !. ES ekrane pateikia:

```
interf_pagalba :-  
  nl, write('-----'), nl,  
  nl, write(' Sistemos komandos sąrašas ir paaiškinimai: '),  
nl, nl,  
  write('      1. "uzkrauti" - bus užkraunamas Jūsų  
pasirinktas žinių bazės failas. '), nl,  
  write('      2. "analizuoti" - bus inicijuojamas pasiekiamų  
būsenų grafo verifikavimo procesas. '), nl,  
  write('      3. "baigti" - bus baigtas sistemos darbas. '),  
nl,  
  nl, write('-----'),
```

Tai suteikia sistemos vartotojui daugiau informacijos apie sistema. Ekrane atsiranda pagrindinis sakinytis kuris paaiškina, kad yra trys komandos. Jei vartotojas pasirenka komanda "uzkrauti" tai reiškia, kad vartotojas iš žinių bazės užkrauną norimą failą. Jei pasirenka "analizuoti" yra analizuojama failas, o jei įvedama komanda "baigti" ES baigia darbą. Pasirinkus komandą užkrauti:

```
uzkrauti_zb :-  
  nl, write('-----'),  
nl,  
  nl, write('Viengubose kabutėse įveskite žinių bazės failo  
pavadinimą (pvz.: analize.pro): '), nl,  
  nl, write('-----'),  
nl,  
  nl, write('> '),  
  read(F),  
  reconsult(F),  
  nl, write('-----'),  
nl,
```

Įvedus reikiamo failo pavadinimą ES pradeda vykdyti verslo procesų analizę.

```
nl, write('Žinių bazės failas užkrautas, galite pradėti  
verslo procesų specifikacijos analizę. '), nl,  
write('Tai atlieka komanda "analizuoti" . '), nl,
```

```

    nl, write('Komanda "pagalba" Jums suteiks papildomą
informaciją. '), nl,
    nl, write('-----'),
nl, nl.

```

Komanda analizuoti pradeda analizuoti pasirinktą failą iš žinių bazės.

```

analizuoti :-
    nl, write('-----'),
nl,
    nl, write('Atsakymas nerastas. '), nl,
    nl, write('Komanda "pagalba" Jums suteiks papildomos
informacijos. '), nl,
    nl, write('-----'),
nl, nl.

```

Taip yra išanalizuojamas pasirinktas verslo procesas, ES ekrane pateikiama informacija apie verslo procese esančias klaidas, aklavietės, ciklus. Parašomi visi rasti galimi necikliniai keliai, taip pat para

3.5. Trečiojo skyriaus išvados

1. Verslo procesų modeliavimas yra sudėtingas ir daug klaidų apimantis procesas, nes klaidos gali būti paliktos modelio specifikacijoje.

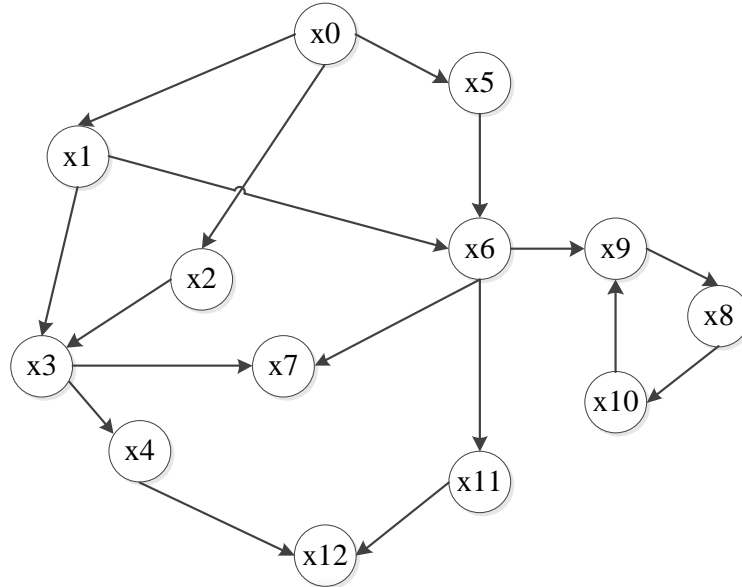
2. Deja, dažniausiai naudojami verslo procesų modeliai, pavyzdžiui, UML, BPMN, Petri tinklai yra skirti verslo procesų dokumentavimui (aprašymui), bet nėra skirti tiesiogiai vykdyti. Minėtos notacijos turi būti transformuojamos į vykdomas programavimo kalbas. Tai reikalauja didelio programuotojų darbo.

3. Dirbtinio intelekto Prolog kalba ir joje realizuotas išvedimo mechanizmas, pagrįstas paieškos algoritmais (paieška gilyn), leidžia labai nesudėtingai aprašyti ir analizuoti grafus. Darbe realizuotas modifikuotas paieškos algoritmas leidžia aptikti aklavietes ir neefektyvius ciklus.

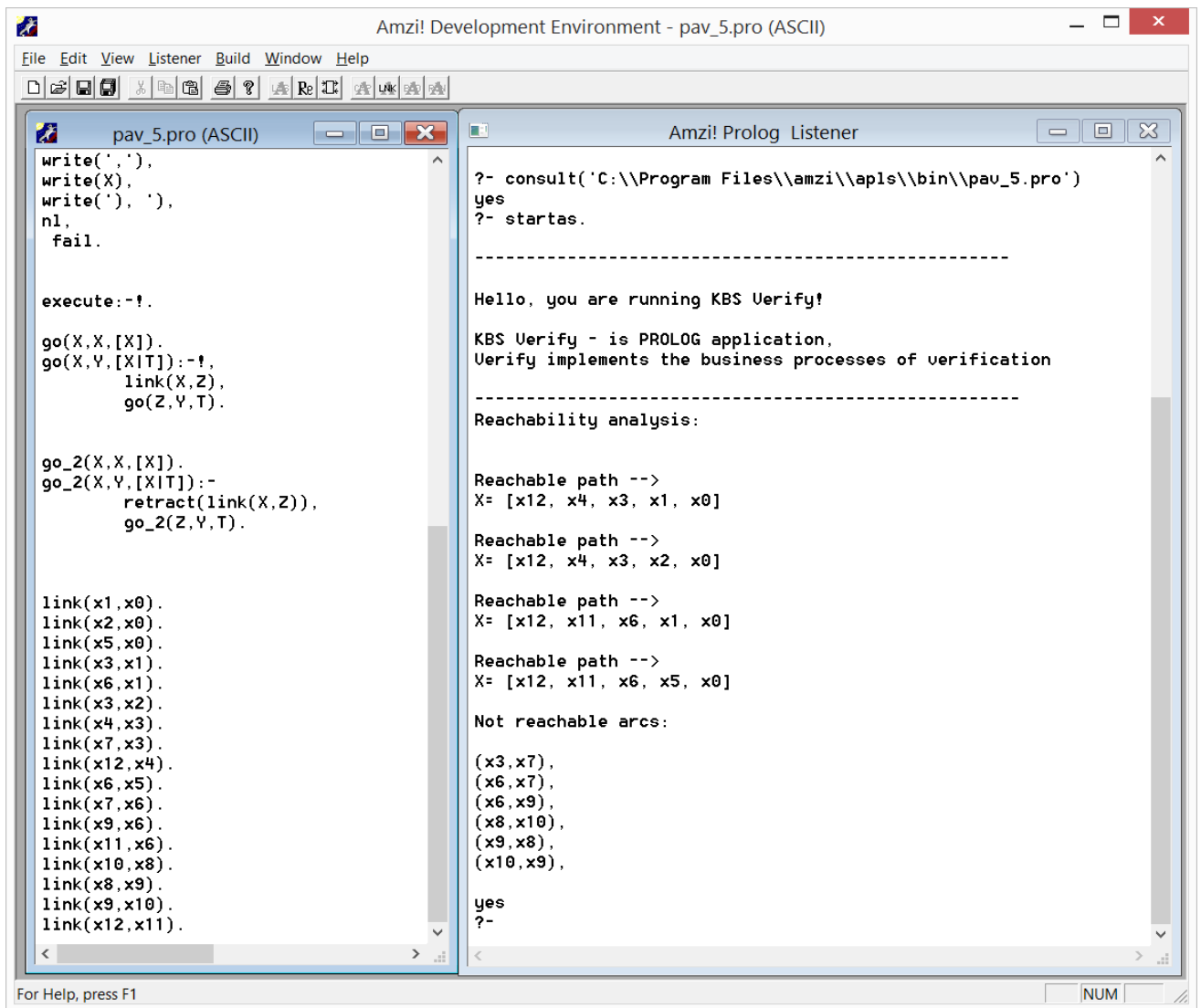
4. EKSPERIMENTAI

4.1. Verslo proceso verifikavimo pavyzdys

Pateikiame verslo proceso modeliavimui ir verifikavimui Prolog kalba analizės rezultatus (30-31 pav.) pavyzdžio, kuris buvo aprašytas ankstesniame skyriuje (6 pav.).



30 pav. Grafo pavyzdys [9]



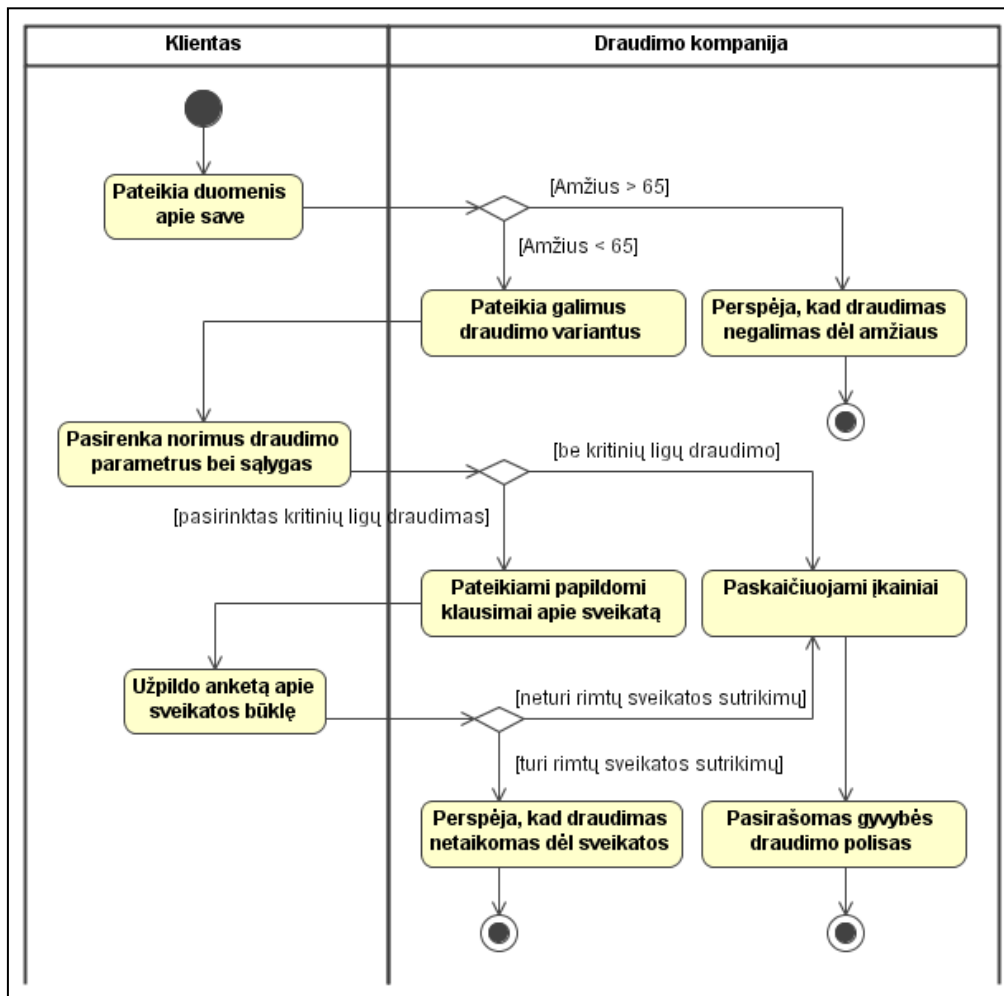
31 pav. Ekspertinės sistemos Prolog rezultatai

Iš rezultatų matyti, kad grafo lankai, vedantys į aklavietes ir neefektyvius ciklus yra:

- (x3,x7),
- (x6,x7),
- (x6,x9),
- (x8,x10),
- (x9,x8),
- (x10,x9).

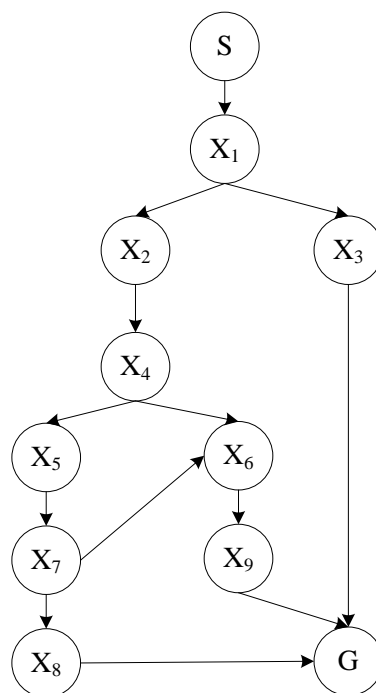
4.2. Gyvybės draudimo verslo proceso verifikavimas

Sekantis eksperimentinis pavyzdys yra - gyvybės draudimo procesas. Pirmiausia pateikiame sveikatos draudimo aprašymą UML veiklos diagrama. Kurioje yra akivaizdžiai pavaizduota kaip vyksta visas procesas: kai žmogus pateikia informaciją apie save draudimo kompanijai, kur draudimo kompanijos atstovas peržiūri pateiktą informaciją ir pagal įmonės vidines taisykles nustato, ar pateiktas prašymas galimas ar ne (kadangi yra žmogaus metų limitas, kuris negali būti viršytas jei norima apsidrausti sveikata). Ir taip nuosekliai pavaizduota visi veiksmai iki paskutinio veiksmo - pabaigos, kaip pateikta (32 pav.).

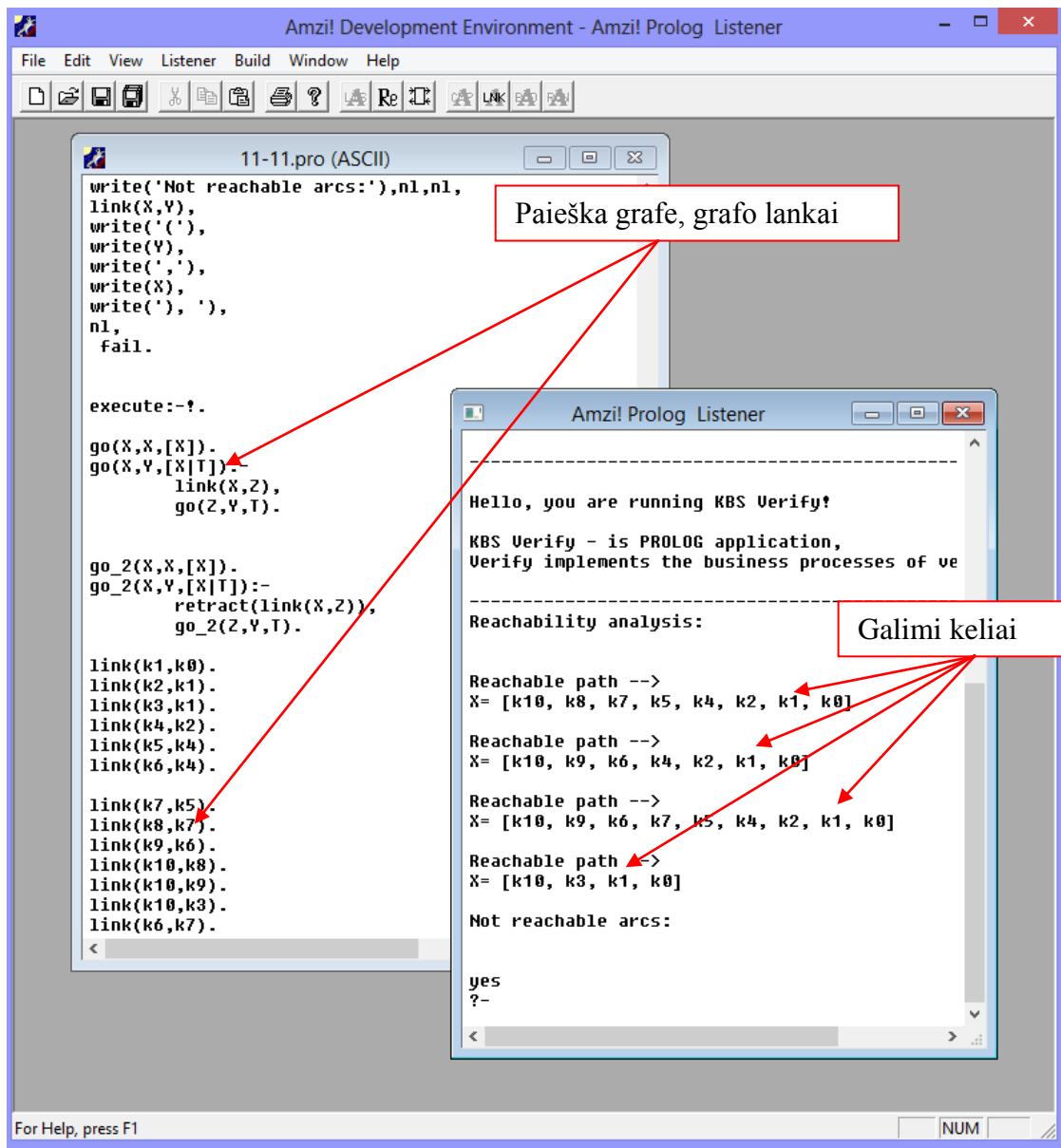


32 pav. Gyvybės draudimo UML veiklos diagrama [9]

Analizuodami toliau šį gyvybės draudimo procesą jį pavaizduojame į orientuotą grafą (33 pav.)

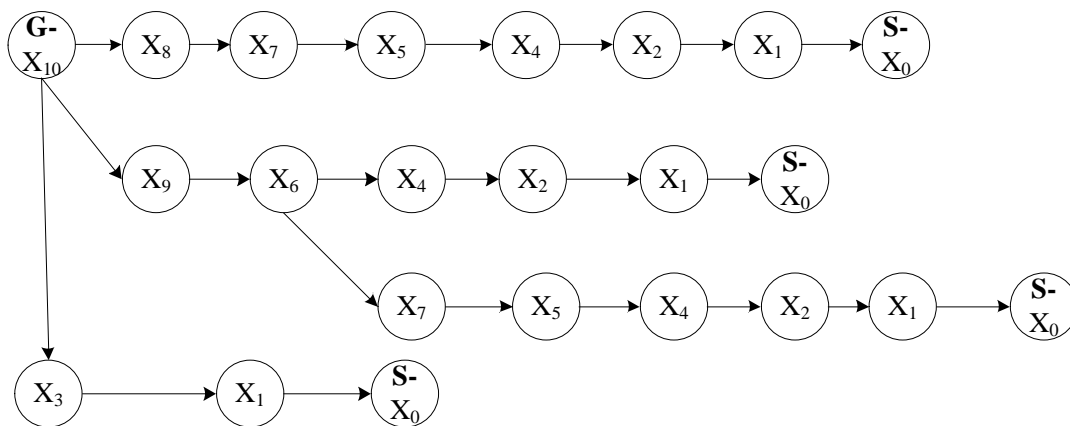


33 pav. Gyvybės draudimo proceso grafas



34 pav. Gyvybės draudimo proceso realizavimas su Prolog kalba

Kaip pateikta (34 pav.) matyti, kad surasti visi galimi šio proceso keliai, iš gautų rezultatų matyti, kad procesas neturi nei aklaviečių, nei ciklų ir bus įvykdomas iki galutinės viršūnės. Analogišką sprendimą gautume grafa atvaizduojant į medį (35 pav.)



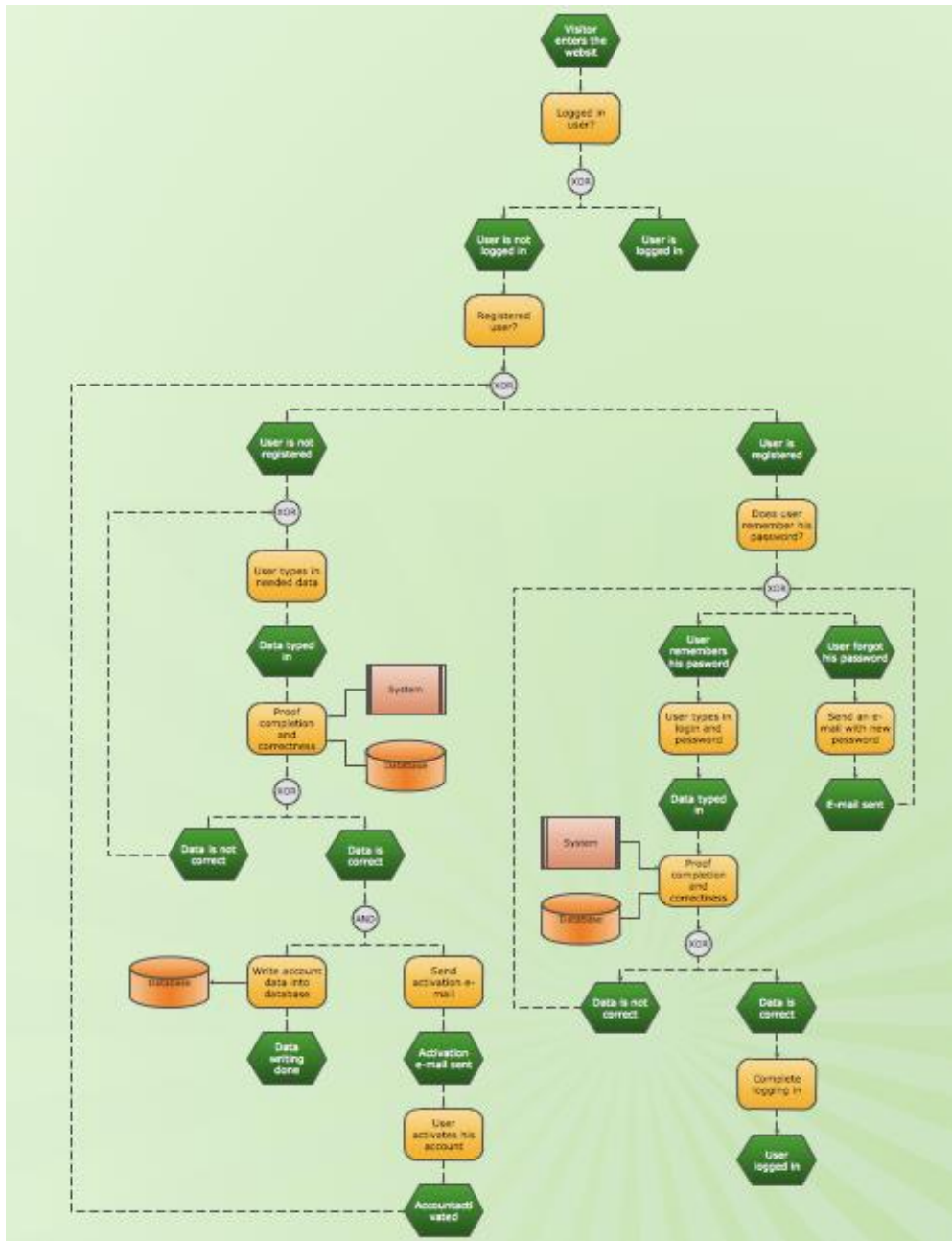
35 pav. Gyvybės draudimo proceso sprendimo medis

4.3. Internetinės svetainės registracijos proceso verifikavimas

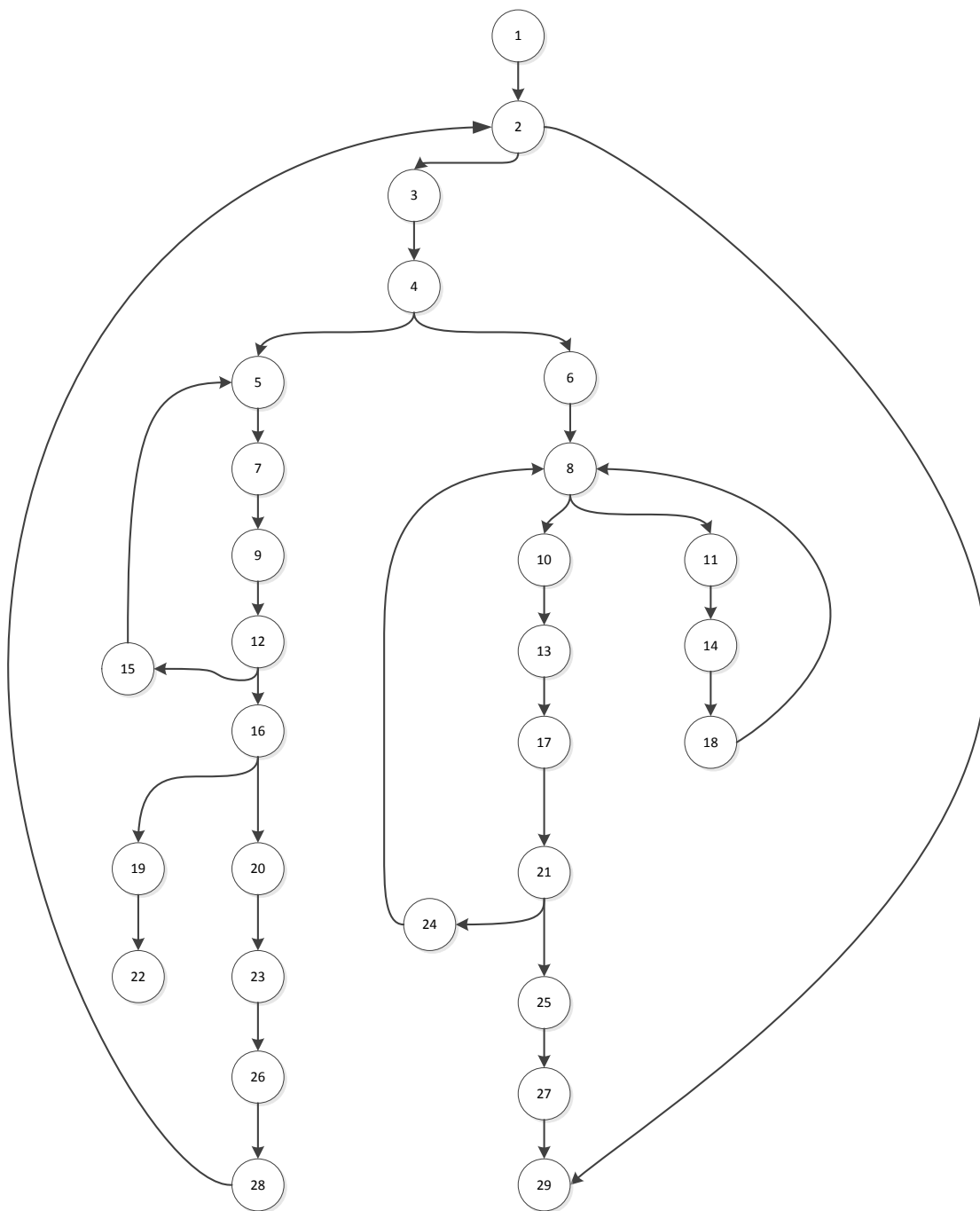
Sekantis nagrinėjamas pavyzdys yra apie internetinės svetainės prisijungimo ir prisiregistravimo procesą (36 pav.). Pateiktoje diagramoje yra pavaizduota kaip lankytojas prisijungia prie pasirinktos svetainės. Įmanomi du variantai arba lankytojas sėkmingai prisijungia prie sistemos arba neprisijungia. Jeigu vartotojas neprisijungia prie svetainės yra du variantai, vartotojas yra užsiregistravęs, bet pamiršo slaptažodį, arba sistemoje dar neprisiregistravęs. Šiuo atveju, kai vartotojas neužsiregistravęs sistemoje, jis turi užpildyti reikalingus registracijos duomenis, kuriuos patvirtina sistemos duomenų bazė. Jeigu vartotojas sėkmingai užpildė pateiktus registracijos laukelių duomenis ir neiškyla problemų, vartotojas yra užregistruojamas. Naujas vartotojas gauną iš svetainės laišką, kuriame yra pateikta sėkminga registracija, kai tai įvykdžius registracija baigta.

Priešingu atveju, kai vartotojas jau yra užsiregistravęs sistemoje, bet negali prisijungti savo duomenimis. Galimi du variantai, vartotojas neatsimena savo slaptažodžio, vartotojas sėkmingai suvedė savo duomenis ir prisijungė. Bet jeigu vartotojas pamiršo slaptažodį, tuomet jis turi parašyti laišką sistemos administratoriui, kuris atsiųs laišką su nauju vartotojo slaptažodžiu. Gavęs šį laišką vartotojas gali vėl jungtis prie sistemos. Tuo atveju jei vėl iškyla problemų su slaptažodžiu, arba bet kokiais kitais duomenų neatitikimais situacija kartojasi analogiškai esančiai prieš tai.

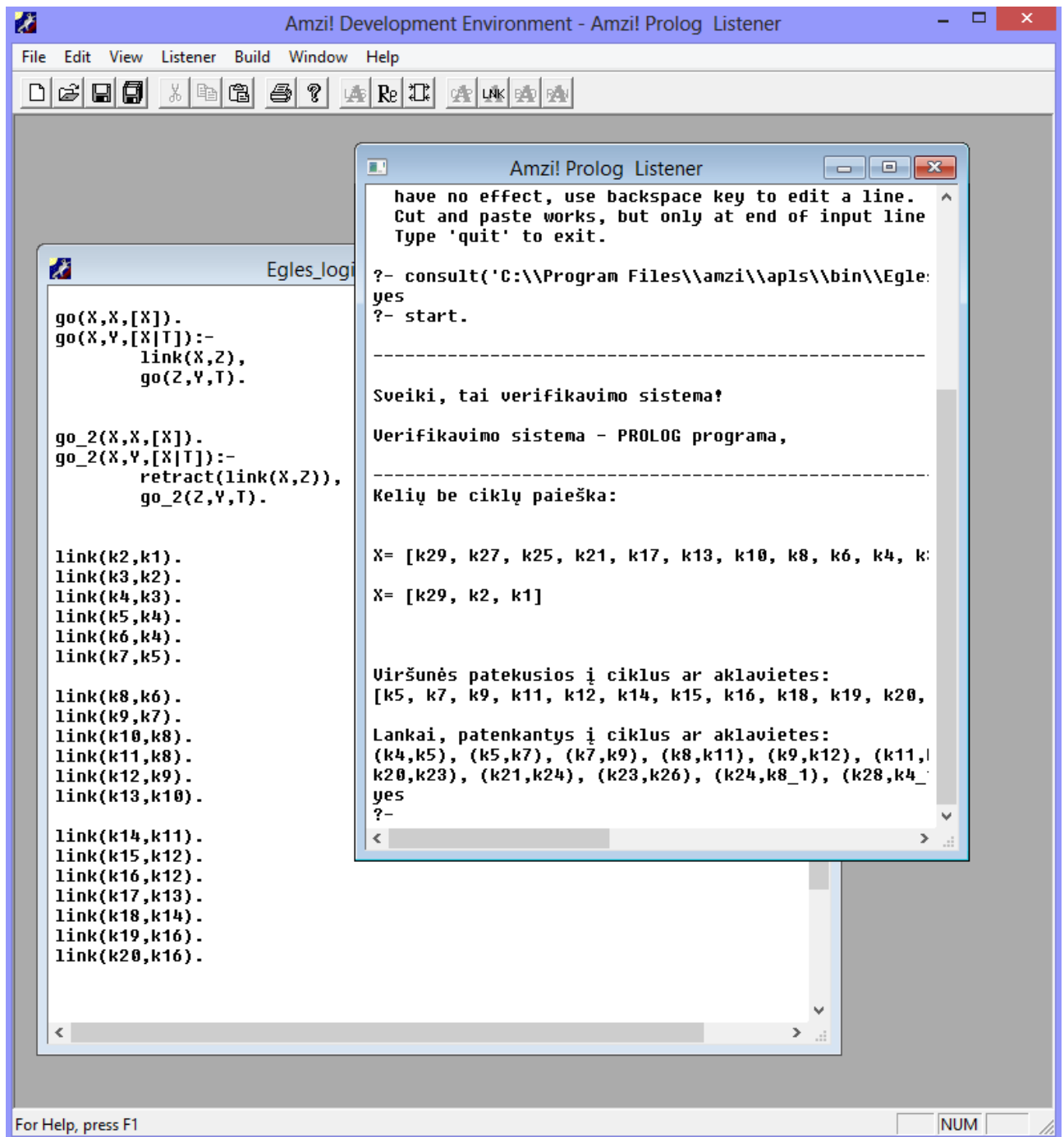
Kaip, kad prieš tai nagrinėtą veiklos proceso diagramą atvaizdavome į orientuotą grafą taip ir šį pavyzdį pervaizduojame/pavaizduojame į grafą (37 pav.). Tik neatvaizduojame tokių laukų, kaip duomenų bazė ar sistema, nes jie jokios įtakos neturi atvaizduojamajame grafe.



36 pav. Internetinės svetainės prisijungimo proceso veiklos diagrama [16] [33]



37 pav. Internetinės svetainės veikimo grafas



38 pav. Internetinės svetainės prisijungimo realizavimas su Prolog

Šiame veiklos procese, kuris buvo tikrinamas Prolog kalba, kaip pateikta (38 pav.), yra matoma, kad procese atsiranda ciklų ir aklaviečių. Taip pat randami du galimi keliai kurie pasiekia veiklos proceso pabaigą.

4.4. Metinės pajamų mokesčių deklaracijos užpildymo proceso verifikavimas

Sekantis nagrinėtas verslo procesas yra susietas su metine pajamų deklaracija. Metinė mokesčių deklaracija – tai mokesčių mokėtojo pateikiamas mokesčių dokumentai, kuriame nurodoma informacija apie mokesčių mokėtojo apskaičiuotą mokesčio sumą už mokesčio įstatyme nustatytą laikotarpį, taip pat su mokesčio apskaičiavimu susiję kiti duomenys. Pagal mokesčių rūšis gali būti Gyventojų pajamų mokesčio deklaracija, Pelno mokesčio deklaracija ir t. t. [34].

Pildydami metinę pajamų deklaraciją žmonės padaro daug klaidų ir taip pateikia klaidingas mokesčių deklaracijos formas. Tai nulemia, kad žmonės nežino kur kokias pajamas įrašyti arba blogai apskaičiuoja tam tikrus laukelius.

Norėdami išvengti tokių klaidų kiekvienas žmogus turėtų gerai išmanyti mokesčių deklaracijos pildymo formas ir jų priedus. Kadangi mokesčių deklaracija žmonės pildo tik kartą į metus todėl nėra linkę labai aiškintis ir analizuoti deklaracijų pildymo taisykles.

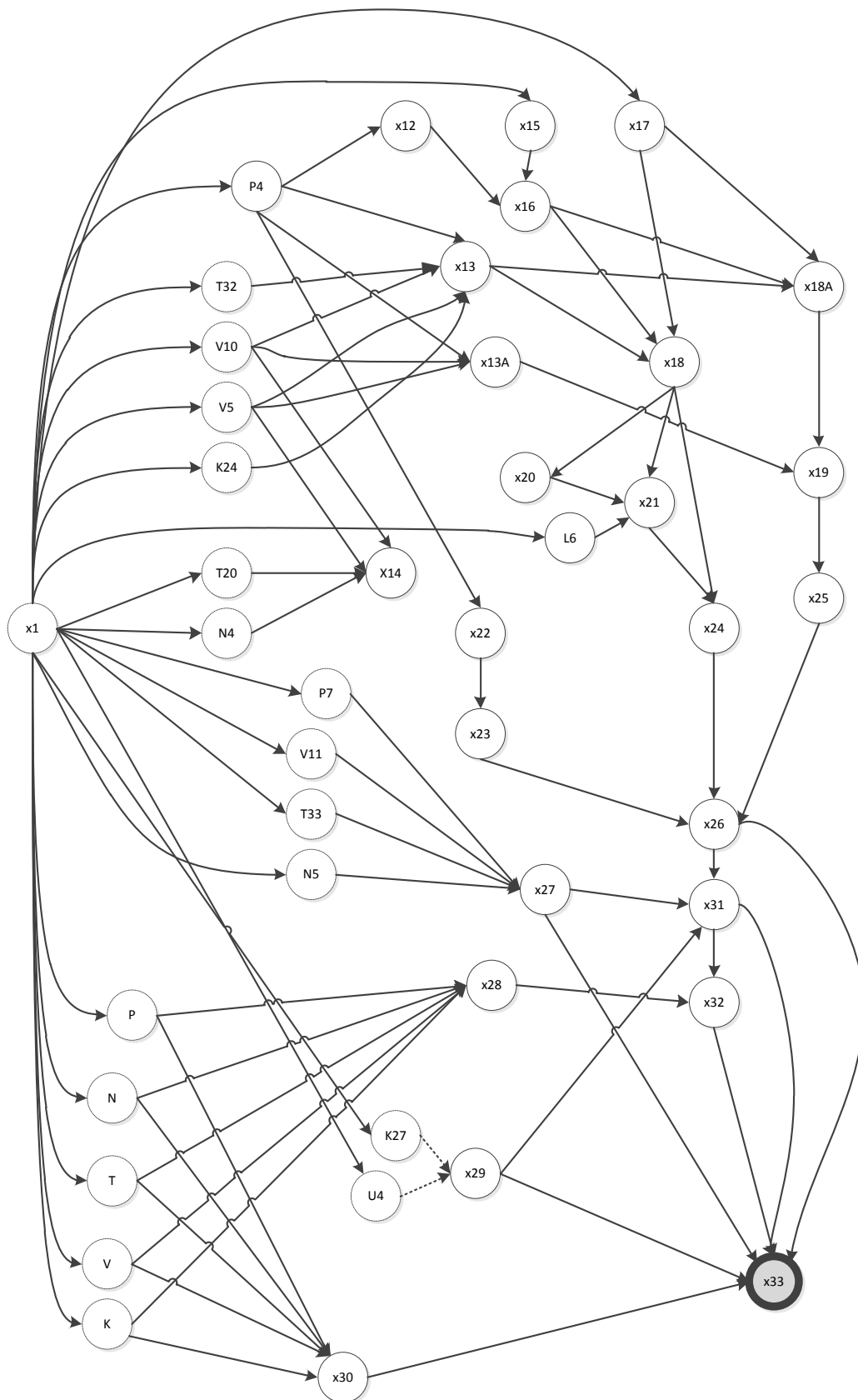
Bandoma išanalizuoti pagrindinę metinę pajamų deklaracijos forma GPM 308 ir ją atvaizduoti į grafą, kurį išnagrinėsime Prologu. Taip siekdami įsitikinti ar metinėse pajamų deklaracijos formose nėra kokių neatitikimų ar aklaviečių.

Pagal turimą metinę pajamų deklaracijos formą GPM 308 sudaroma lentelė kurioje atsispindi visi formos laukai, aprašymai ir formulės (2 lentelė). Verifikavimo rezultatai pateikti 39 ir 41 paveiksluose.

2 lentelė. Metinės pajamų deklaracijos formos GPM 308 formos laukai

Žymėjimo numeris	Laukelio formos paaiškinimas	Laukelio sudarymo formulės
6	Mokestinis laikotarpis	
7	Iki (pildo tik išvykstantysis)	
8	Galutinai išvykstančio iš Lietuvos nuolatinio Lietuvos gyventojų	
9	Pateikiami priedai	GPM308P GPM308L GPM308N GPM308K GPM308T GPM308V GPM308R GPM308U GPM308S
10	Deklaracijos lapų skaičius	
11	Pridedamų įgaliojimo, užsienio valstybėje išduotų dokumentų, FR0781 formos prašymo lapų skaičius	
12	Su darbo santykiais susijusios pajamos (Lt, ct)	(P4 [kai P2 yra 01, 02 ir 03])
13	Taikant 15 proc. pajamų mokesčio tarifą apmokestinamos pajamos (Lt, ct)	(P4 [kai P1 yra 15, išskyrus, kai P2 yra 01, 02 ar 03] +T32 + V10 [kai V5 yra 15] + K24)
13A	Taikant 5 (10) proc. pajamų mokesčio tarifą apmokestinamos pajamos (Lt, ct)	(V10 [kai V5 yra 5 arba 10] + P4 [kai P1 yra 5 arba 10])
14	Neapmokestinamosios pajamos (Lt, ct)	(N4+T20+V10 [kai V5 yra 0])
15	Taikytino MNPD suma (Lt, ct)	
15A	Mėnesių skaičius*	
15B	Mėnesių skaičius**	
16	Su darbo santykiais susijusios pajamos, atėmus MNPD (Lt, ct)	
17	Taikytino MPNPD suma (Lt, ct)	
18	Taikant 15 proc. pajamų mokesčio tarifą apmokestinamų pajamų suma, atėmus MPNPD (Lt, ct)	(13+16 - 17) (jei gaunama neigiama suma, įrašomas nulis)
18A	MPNPD suma, neatimta iš taikant 15 proc. mokesčio tarifą apmokestinamų pajamų	(17-13 -16) (jei gaunama neigiama suma, įrašomas nulis)
19	Taikant 5 (10) proc. pajamų mokesčio tarifą apmokestinamos pajamos, atėmus neatimtą MPNPD (Lt, ct)	(13A - 18A) (jei gaunama neigiama suma, įrašomas nulis)
20	20 ir 21 laukeliai pildomi, kai užpildytas	(18 x 0,25)

	GPM308L priedas Maksimali apmokestinamąsias pajamas mažinanti suma (Lt, ct)	
21	Taikant 15 proc. mokesčio tarifą apmokestinamų pajamų suma, atėmus išlaidas (Lt, ct)	(18-20, kai visų lapų L6>20 arba 18 - visų lapų L6, kai L6<=20)
22	Taikant 20 proc. pajamų mokesčio tarifą apmokestinamų pajamų suma (Lt, ct)	(P4 [kai P1 yra 20])
23	Pajamų mokesčio suma nuo pajamų, apmokestinamų taikant 20 proc. tarifą (Lt)	(22 lauk. x 0,2)
24	Pajamų mokesčio suma nuo pajamų, apmokestinamų taikant 15 proc. tarifą (Lt)	(21 lauk. x 0,15 arba 18 lauk. x 0,15)
25	Pajamų mokesčio suma nuo pajamų, apmokestinamų taikant 5 (10) proc. tarifą (Lt)	(19 lauk. x 0,05 (0,1))
26	Bendra apskaičiuota pajamų mokesčio suma (Lt)	(23 +24+ 25)
27	Bendra išskaičiuota pajamų mokesčio suma (Lt)	(P7 + V11 + T33 + N5)
28	Bendra kito asmens savo lėšomis sumokėta pajamų mokesčio suma (Lt)	(nuo pajamų, nurodytų P, N, T, V, K prieduose)
29	Atskaitoma užsienio valstybėse išskaičiuoto (sumokėto) pajamų mokesčio suma (Lt)	(U4 + K27)
30	Bendra gyventojų savo lėšomis iki deklaracijos pateikimo dienos sumokėta pajamų mokesčio suma (Lt)	(nuo pajamų, nurodytų P, N, T, V, K prieduose)
31	Gyventojų mokestinė prievolė (+/-), neatsižvelgus į kito asmens sumokėtą pajamų mokestį (Lt)	(26 - 27 - 29)
32	Gyventojų mokestinė prievolė, sumažinta kito asmens savo lėšomis sumokėtu mokesčiu (Lt)	31-28; pildoma, kai 31 lauk.>0 jei gaunama neigiama suma, įrašomas nulis)
33	Mokėtina (-) / grąžintina (+), pajamų mokesčio suma (Lt)	(30 - 32 , kai 31>0 30 + 27 + 29 - 26, kai 31 <=0)



39 pav. Metinės pajamų mokesčio deklaracijos grafas


```

Amzi! Prolog Listener
Version: 9.5.2 64-bit
Copyright (c) 1992-2012 Amzi! inc. All Rights Reserved.

*****
Amzi! Prolog (IDE Only)
Free Version
*****

yes
Listener 'listens to' input from keyboard,
not screen. Multiline input is OK, '.' followed
by 'Enter' enters line. Mouse and cursor keys
have no effect, use backspace key to edit a line.
Cut and paste works, but only at end of input line.
Type 'quit' to exit.

?- consult('C:\\Program Files\\amzi\\apls\\bin\\deklaracija.pro')
yes
?- startas.

-----
Sueiki, tai verifikavimo sistema!

Verifikavimo sistema - PROLOG programa.

-----
Kelių be ciklų paieška:

X= [x33, x26, x23, x22, x1]
X= [x33, x26, x24, x21, x20, x18, x13, x1]
X= [x33, x26, x24, x21, x20, x18, x16, x12, x1]
X= [x33, x26, x24, x21, x20, x18, x16, x15, x1]
X= [x33, x26, x24, x21, x20, x18, x17, x1]
X= [x33, x26, x24, x21, x1]

X= [x33, x26, x24, x21, x18, x13, x1]
X= [x33, x26, x24, x21, x18, x16, x12, x1]
X= [x33, x26, x24, x21, x18, x16, x15, x1]
X= [x33, x26, x24, x21, x18, x17, x1]
X= [x33, x26, x24, x18, x13, x1]
X= [x33, x26, x24, x18, x16, x12, x1]
X= [x33, x26, x24, x18, x16, x15, x1]
X= [x33, x26, x24, x18, x17, x1]
X= [x33, x26, x25, x19, x13A, x1]
X= [x33, x26, x25, x19, x18A, x16, x12, x1]
X= [x33, x26, x25, x19, x18A, x16, x15, x1]
X= [x33, x26, x25, x19, x18A, x17, x1]
X= [x33, x26, x25, x19, x18A, x13, x1]
X= [x33, x32, x31, x29, x1]
X= [x33, x32, x31, x27, x1]
X= [x33, x32, x31, x26, x23, x22, x1]
X= [x33, x32, x31, x26, x24, x21, x20, x18, x13, x1]
X= [x33, x32, x31, x26, x24, x21, x20, x18, x16, x12, x1]
X= [x33, x32, x31, x26, x24, x21, x20, x18, x16, x15, x1]
X= [x33, x32, x31, x26, x24, x21, x20, x18, x17, x1]
X= [x33, x32, x31, x26, x24, x21, x1]

```

40 pav. Metinės pajamų mokesčio deklaracijos verifikavimas su Prolog

```

X= [x33, x31, x26, x24, x21, x18, x16, x15, x1]
X= [x33, x31, x26, x24, x21, x18, x17, x1]
X= [x33, x31, x26, x24, x18, x13, x1]
X= [x33, x31, x26, x24, x18, x16, x12, x1]
X= [x33, x31, x26, x24, x18, x16, x15, x1]
X= [x33, x31, x26, x24, x18, x17, x1]
X= [x33, x31, x26, x25, x19, x13A, x1]
X= [x33, x31, x26, x25, x19, x18A, x16, x12, x1]
X= [x33, x31, x26, x25, x19, x18A, x16, x15, x1]
X= [x33, x31, x26, x25, x19, x18A, x17, x1]
X= [x33, x31, x26, x25, x19, x18A, x13, x1]

Viršūnės patekusios į ciklus ar aklavietes:
[x14]

Lankai, patenkantys į ciklus ar aklavietes:
(x1,x14),
yes
?-

```

41 pav. Metinės pajamų mokesčio deklaracijos verifikavimas su Prolog (tęsinys)

Patikrinus pasirinktą pavyzdį su sukurta EK buvo gautas rezultatas kuriame atsispindi galimi keliai kurie pasiekia pradinę viršūnę. Tokių kelių buvo ne vienas, galima teikti, kad deklaracijos procesų elementai eina nuosekliai ir lygiagrečiai, kurie nekelia grėsmės, kad atsirastų aklaviečių. Taip pat yra pateikiama informacija ir apie lankus ar viršūnes kurios galimai gali sukelti problemų. Tokiu atveju reikėtų atidžiau panagrinėti tas viršūnes ar lankus kuriuos surado ES ir pateikė į rezultatų ekraną. Nereikėtų daryti prielaidos, kad būtinai rasti lankai ar viršūnės sukels verslo procesui problemų, kad jis nepasieks pabaigos.

Paveiksluose (40 pav.) ir (41 pav.) buvo panaudotos tik pagrindinė komanda "analizuoti", kuri į ES ekraną išvedė rezultatus. Paveiksluose (42 pav.) pateikti rezultatai kaip pasirinkama komanda "interf_pagalba.". Matoma, kad ekspertinė sistema į ekraną išvedė tris komandas: uzkrauti, analizuoti, baigti.

```
-----  
yesreconsult('C:\\Users\\egle\\Desktop\\prolog inter\\Egles_interfeisas.pro')  
yes  
?- interf_pagalba.  
-----  
  
Sistemos komandų sąrašas ir paaiškinimai:  
  
1. "uzkrauti" - bus užkraunamas Jūsų pasirinktas žinių bazės failas.  
2. "analizuoti" - bus inicijuojamas pasiekiamumo būsenos grafo verifikavimo procesas.  
3. "baigti" - bus baigtas sistemos darbas.  
-----  
  
yes  
?-  
?-
```

42 pav. Komandos *interf_pagalba* rezultatai

Iš pateiktų komandų pasirenkama komanda "uzkrauti_zb", ES ekrane yra pateikiama tokia informacija kaip pateikta (43 pav.), kuris praneša vartotojui, kad nurodytą failo pavadinimą, kuris bus analizuojamas.

```
?- uzkrauti_zb.  
-----  
Viengubose kabutėse įveskite žinių bazės failo pavadinimą (pvz.: analize.pro):  
-----  
  
> deklaracija.  
-----  
Žinių bazės failas užkrautas, galite pradėti verslo procesų specifikacijos analizę.  
Tai atlieka komanda "analizuoti" .  
Komanda "pagalba" Jums suteiks papildomą informaciją.  
-----  
  
yes  
?-  
?-
```

43 pav. Komandos *uzkrauti* rezultatai

Pateikus gerą failo pavadinimą jis yra užkraunamas, pranešama, kad pasirinktas failas yra žinių bazėje ir jį galima analizuoti. Pateikus komanda analizuoti yra gaunami tokie rezultatai kaip pateikta paveiksluose (40 pav.) ir (41 pav.).

4.5. Ketvirto skyriaus rezultatai ir išvados

Atliekant eksperimentinę dalį buvo analizuojami keli verslo procesai, tokie kaip: pavyzdinis orientuotas grafas su klaidomis, gyvybės draudimo verslo procesas, internetinės svetainės registracijos procesas ir metinės pajamų/ mokesčių deklaracijos užpildymo procesas.

Pirmasis eksperimentas buvo atliekamas su paprastu orientuotu grafu, kuriame buvo suimituota aklavietė. Grafo lankai buvo pateikti į sukurtą ekspertinę sistemą (ES), kuri surado ir perspėjo apie esamą aklavietę, taip pat pranešė ir apie neefektyvius ciklus.

Sekantis eksperimentas buvo atliktas su gyvybės draudimo procesu, kuris buvo pavaizduotas UML veiklos diagrama. Pagal darbe pateiktą metodiką, ši diagrama buvo pavaizduota į orientuotą grafą. Šio grafo lankai kaip duomenys verifikavimui buvo įkelti į ekspertinę sistemą. ES pateikė rezultatus, kuriuose matyti, kad gyvybės draudimo procese nėra nei aklaviečių nei ciklų.

Kitas eksperimentas jau buvo atliekamas sudėtingesnis, norint įsitikinti ar sukurta ES veikia teisingai ir nepriklauso nuo lankų ar būsenų kiekio ir geba verifikuoti didesnius verslo procesus. Eksperimentui pasirinktas verslo pavyzdys susietas su internetinės svetainės prisijungimu. Pasirinktas procesas, kuris buvo literatūroje pateiktas srauto diagrama, pirmiausia buvo pavaizduotas orientuotu grafu. Iš srauto diagramos buvo pašalintos tokios būsenos kaip duomenų bazės, saugyklos, kurios jokios įtakos neturi proceso verifikavimui. Į ES parašome būsenas ir lankus taip pradėdami verifikavimą. Ekspertinė sistema parodė, kad procese yra surandami du keliai be ciklų ir aklaviečių. Tai, kad ES pateikia rezultatuose viršūnes ar lankus kurie patenka į aklavietes, ciklus ir t.t tai nebūtinai reiškia, kad procesas nebus įgyvendintas. Ekspertinė sistema tik vartotoją perspėja, kad tokie veiksmai, darbai ar jų seka gali turėti kažkokių kliūčių todėl į jas reikėtų atkreipti didesnę dėmesį įgyvendinant verslo procesą.

Paskutiniam eksperimentui buvo pasirinktas metinės pajamų mokesčių deklaracijos užpildymo procesas. Buvo analizuojama forma GPM 308. Apžvelgus pasirinktą formą buvo sudaryta lentelė, kurioje pateikti visi šios deklaracijos laukai ir skaičiavimo formulės. Pagal darbe pateiktą metodiką, deklaracija buvo pavaizduota į orientuotą grafą. Deklaracijos formos grafe yra atvaizduoti visi laukai kurie yra ir formoje, bet tik priedai sujungti į vieną viršūnę x1. Verifikavimo rezultatai ekspertinėje sistemoje parodė, kad deklaracijos formos nuoseklume ir pildymo eigoje yra galima aklavietė. Taip pat pateikti ir visi viršūnių keliai, kuriuose nėra ciklų ir aklaviečių.

5. IŠVADOS

1. Atlikta mokslinių šaltinių analizė parodė, kad šiuo metu verslo procesų logikai aprašyti naudojami metodai (tokie kaip verslo procesų modeliavimo notacija (BPMN), unifikauta modeliavimo kalba (UML veiklos diagramos) bei kiti formalizmai) yra gana sudėtingi ir, verslo atstovams, kurie nežino minėtų formalių procesų užrašymo būdų, sunku susišnekėti su informacinių technologijų specialistais modeliuojančiais šiuos procesus. Be to, BPMN, UML veiklos diagramos ir kiti formalizmai naudoja sudėtingas programines sistemas, kad būtų galima patikrinti verslo procesų teisingumą.

2. Šiame darbe pasiūlyta verslo procesų modeliavimui panaudoti grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose. Šio metodo privalumas toks, kad jis leidžia grafiškai aprašyti nesudėtingą procesų logiką, o tokia grafų forma gerai suprantama tiek verslo atstovams, tiek ir informacinių technologijų specialistams.

3. Verslo procesų verifikavimui sukurta ekspertinė sistema, kuri realizuota Prolog kalba. Dirbtinio intelekto Prolog kalboje realizuotas išvedimo mechanizmas, pagrįstas paieškos algoritmais (paieška gilyn), leidžia labai nesudėtingai aprašyti ir analizuoti grafus. Darbe realizuotas modifikuotas paieškos algoritmas leidžia aptikti aklavietes ir neefektyvius ciklus.

6. LITERATŪROS SĄRAŠAS

1. Business Process Model and Notation (BPMN) version 2.0 [Tinkle]. January, 2011; p. 1-41. Prieiga per internetą: <http://www.omg.org/spec/BPMN/2.0/PDF>. [kreiptasi 2013-01-10]
2. M. Weske, Business Process Management: Concepts, Languages, Architectures [Tinkle]. November, 2007; p. 1-21. Prieiga per internetą: http://books.google.lt/books?id=QMyu_B1KTZIC&redir_esc=y. [kreiptasi 2013-02-06]
3. R. G. Ross, How Rules and Processes Relate. Part 2. Business Processes. In Business Rules Journal, Vol. 6, No. 11. Prieiga per internetą: <http://www.brcommunity.com/b256.php>. [kreiptasi 2013-02-15]
4. S. Gudas, A. Lopata, Darbų sekų modeliais grindžiamas veiklos žinių surinkimo būdas. p. 277-278, 2005.
5. D. Šilingas, A. Laugalis, *Verslo procesų valdymas: kas tai, kodėl ir kaip?* [Tinkle]. Prieiga per internetą: <http://www.vpvp.lt/straipsniai>. [kreiptasi 2013-03-25]
6. *Verslo procesų valdymas* [Tinkle]. Prieiga per internetą: <http://www.sprendimuidejos.lt/index.php/paslaugos/proces-valdymas>. [kreiptasi 2013-03-25]
7. *Įmonės procesų žemėlapis* [Tinkle]. Prieiga per internetą: <http://estrategija.lt/23ProcesuZemelapis/ProcesuZemelapis.htm>. [kreiptasi 2013-03-27]
8. A. Pautienius, *ES struktūrinės paramos projektų efektyvumas* [Tinkle]. Prieiga per internetą: <http://is.mokslasplus.lt/wp-content/uploads/2013/05/ES-PARAMOS-PROJEKT%C5%B2-EFEKTYVUMAS-I.pdf>. [kreiptasi 2013-03-27]
9. R. Misevičienė, Č. Christauskas, E. Baležentytė, Verslo procesų verifikavimas, panaudojant modifikuotą paieškos algoritmą [Tinkle]. 2013; p.38-45.]. Prieiga per internetą: <http://www.zara.lt/e-knygos/e-KoDi-2013-metod.pdf>. [kreiptasi 2013-11-27]
10. N. Russell, M. P. Wil, H. M. Arthur ter Hofstede, P. Wohed, On the suitability of UML 2.0 activity diagrams for business process modelling. Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling, p.95-104, January 16-19, 2006, Hobart, Australia
11. *Veiklos diagrama* [Tinkle]. Prieiga per internetą: http://lt.wikipedia.org/wiki/Veiklos_diagrama. [kreiptasi 2013-11-22]
12. M. Kisly, Dokumentų valdymo sistemų darbo sekų transformavimas iš BPMN į WF modelį: magistro darbas. Vilnius 2008.
13. *Comparison of Business Process Modeling Notation tools* [Tinkle]. Prieiga per internetą: http://en.wikipedia.org/wiki/Business_Process_Modeling. [kreiptasi 2013-12-04]
14. *Business process modelling* [Tinkle]. Prieiga per internetą: <http://www.businessballs.com/business-process-modelling.htm>. [kreiptasi 2013-12-04]
15. *Algoritmai grafuose* [Tinkle]. Prieiga per internetą: <http://www.techmat.vgtu.lt/konspektai/Algoritmai/Paskaita91.pdf>. [kreiptasi 2013-12-06]
16. H. Pranevičius, R. Misevičienė, Verification of business process workflows, Technological and Economic Development of Economy. 2012, 623-635, DOI: 10.3846/20294913.2012.740517
17. *Paieška į gylį* [Tinkle]. Prieiga per internetą: http://lt.wikipedia.org/wiki/Paieška_į_gylį. [kreiptasi 2013-12-16]
18. *Paieška į plotį* [Tinkle]. Prieiga per internetą: http://lt.wikipedia.org/wiki/Paieška_į_plotį. [kreiptasi 2013-12-16]
19. B.C. Williams, Problem Solving as State Space Search. Slides adapted from: 6.034 Tomas Lozano Perez, Russell and Norvig AIMA. 2004
20. *Formalus verifikavimas* [Tinkle]. Prieiga per internetą: http://lt.wikipedia.org/wiki/Formalus_verifikavimas. [kreiptasi 2013-12-28]
21. A. Preece, Foundation and Application of Knowledge Base Verification. International Journal of Intelligent Systems 9, 2010.
22. H. Pranevičius, R. Ceponyte, Application of Logic Programming Based Language Prolog for Invariant Proving. *Mathematics and Mathematical Modelling: Thesis of Conference "Achievement of Technical Science in Republic"*. Vilnius, Feb. 1-4, Vilnius, 1990.

23. *Pagrindinės tyrinėjimo bei pritaikymo sritys* [Tinkle]. Prieiga per internetą: irma.vtu.lt/linspace/dies1/schedule.nsf/d862e82eafb758368525663c004f385c/08d5fc3718167444c1256d67004643f1?OpenDocument. [kreiptasi 2014-01-09]
24. N. Paliulis, N. Astrauskienė, *Informacinės valdymo sistemos: mokymo knyga*, Vilnius: 2003. p.171-174.
25. R. T. Plant, *Methodologies for the development of knowledge-based systems. The Knowledge Engineering Review, Cambridge University Press 18*. London, p. 47-81, 2003,
26. D. Merritt. *Building Expert Systems in Prolog, Springer-Verlag*. New York Inc, ISBN 0-387-97016-9, 1989.
27. D. Pečiulytė, *Ekspertinė agregatinių specifikacijų analizės sistema: magistro darbas*. Kaunas 2000.
28. *Ekspertinės sistemos* [Tinkle]. Prieiga per internetą: http://lt.wikipedia.org/wiki/Ekspertinė_sistema. [kreiptasi 2014-01-16]
29. V. Čyras, *Dirbtinis intelektas*. Vilnius 2008; p.115-129.
30. V. Barr, *Applications of rule-base coverage measures to expert system evaluation. Knowledge-Based Systems 12.1 (1999): 27-35*.
31. *Visual Prolog Version 5.0: Language Tutorial / Prolog Development Center A/S*. Copenhagen, Denmark, 1996.
32. *State-Space Search* [Tinkle]. Prieiga per internetą: http://www.inf.ed.ac.uk/teaching/courses/aipp/lecture_slides/08_Search.pdf
33. *Business Processes - Flow Charts. Sample 8: Flow Chart - Website Login and Registration Process* [Tinkle]. Prieiga per internetą: <http://www.conceptdraw.com/samples/business-process-diagrams-flow-charts>. [kreiptasi 2014-03-10]
34. *Mokesčio deklaracija* [Tinkle]. Prieiga per internetą: http://lt.wikipedia.org/wiki/Mokes%C4%8Dio_deklaracija [kreiptasi 2014-03-19]

7. PRIEDAI

7.1. Moksliniai rezultatai, publikuoti „Kompiuterininkų dienos 2013“ konferencijos medžiagoje

Verslo procesų verifikavimas, panaudojant modifikuotą paieškos algoritmą

doc., dr. Regina Misevičienė*

Verslo informatikos katedra, Kauno technologijos universitetas

regina.miseviciene@ktu.lt

Studentų g. 56, LT-51424, Kaunas, + 370 37 300375

doc., dr. Česlovas Christauskas*

Verslo administravimo katedra, Kauno technologijos universitetas

ceslovas.christauskas@ktu.lt

K. Donelaičio g. 20, LT-44239, Kaunas, + 370 37 300126

Magistrantė Eglė Baležentytė

Informatikos fakultetas, Kauno technologijos universitetas

Studentų g. 50, LT-51424, Kaunas, + 370 37 300351

egle.balezentyte@stud.ktu.lt

Santrauka. Darbe siūloma praplėsti verslo procesų modeliavimo ir analizės metodus panaudojant procesų modeliavimui IR/ARA grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose. Šio modeliavimo privalumas toks, kad jis leidžia grafiškai aprašyti verslo procesų logiką, o darbe pateiktas modifikuotas paieškos algoritmas padeda spręsti pasiekiamumo problemas: aptikti aklavietes ir ciklus.

Pagrindiniai žodžiai: verslo procesai, verifikavimas, IR/ARBA grafai

ĮVADAS

Kiekvienoje organizacijoje egzistuoja įvairūs verslo procesai. Nagrinėjant šiuos procesus pagal Poterio vertės grandinės modelį, visi procesai yra skirstomi į pagrindinius, valdymo ir pagalbinius procesus. Pagrindiniai procesai - tai procesai, kuriantys vertę klientui ir tiesiogiai susiję su produktu. Tai - projektavimas, gamyba, pardavimas, aptarnavimas. Valdymo procesai susiję su resursų valdymu. Tai - strateginis planavimas, kontrolė, procesų gerinimas. Pagalbiniai procesai skirti aptarnauti pagrindinius ir verslo valdymo procesus (Verslo, 2013). Nors šių procesų rezultate nesukuriama tiesioginė vertė klientui, bet jie reikalingi pagrindinių procesų veiklai užtikrinti.

Įmonės, siekdamos sutrumpinti darbų atlikimo laiką, susiduria su poreikiu pertvarkyti šiuos verslo procesus ar juos atnaujinti. Todėl verslo procesų valdymas tampa sudėtinga įmonės valdymo dalimi (Vadybos, 2013).

Intensyvus informacinių technologijų vystymasis taip pat skatina naujų verslo procesų modeliavimo priemonių kūrimą. Laikui bėgant pasaulinėje rinkoje atsirado produktų skirtų procesų modeliavimui, valdymui, bei vykdymui.

Šiuo metu formaliai verslo procesų logikai aprašyti naudojami gana įvairūs metodai (Business, 2011; Weske, 2007; Ross, 2005; Gudas2005). Tai verslo procesų modeliavimo notacija (BPMN), unifikuota modeliavimo kalba (UML veiklos diagramos), Petri tinklai, verslo taisyklės ir kiti formalizmai.

Modeliuojant verslo procesus pirmiausia turi būti apibrėžiama verslo procesų logika ir po to patikrinama ar šis aprašas yra tikslus ir teisingas (įsitikinti ar nėra nurodyta neteisingų veiksmų). Dažnai verslo atstovams, kurie nežino minėtų formalių procesų užrašymo būdų, sunku susišnekėti su informacinių technologijų specialistais modeliuojančiais šiuos procesus. Ieškoma vis geresnių būdų veiklos sritims aprašyti modeliais, kurie būtų suprantami visiems.

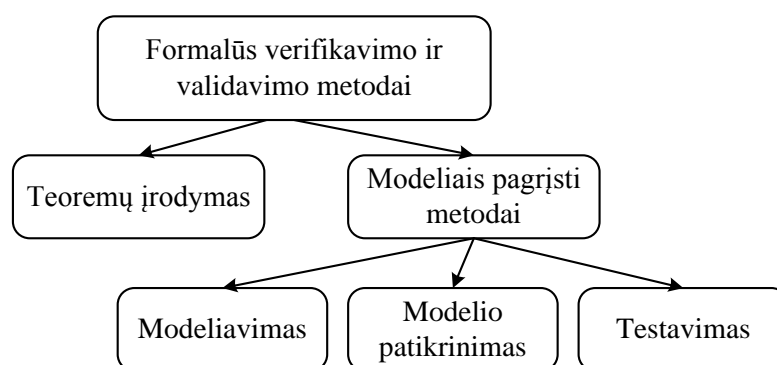
Darbe bus analizuojami tik pagalbiniai verslo procesai. Pagalbiniai procesai yra skirti aptarnauti pagrindinius procesus ir dažniausiai užrašomi darbų atlikimo eigos žodinėmis instrukcijomis arba tiesiog vaizduojami blokinėmis schemomis. Tokių procesų logiką nebūtina aprašyti sudėtingais formaliais metodais tokiais kaip BPMN, UML veiklos diagramomis ar Petri tinklais vien tam, kad vėliau, naudojant sudėtingas programines priemones, būtų galima tikrinti jų teisingumą.

Šiame darbe siūloma praplėsti verslo procesų modeliavimo ir analizės metodus panaudojant procesų modeliavimui IR/ARA grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose. Šių grafų privalumas toks, kad jie leidžia grafiškai aprašyti

nesudėtingą procesų logiką, o tokia grafų forma gerai suprantama tiek verslo atstovams, tiek ir informacinių technologijų specialistams. Yra mokslinių darbų, kuriuose siūloma verslo procesus modeliuoti paprastais orientuotais grafais, tačiau jų teisingumo tikrinimui naudojami ekonometriniai metodai (Jakštonytė, 2006; Pranevicius, 2012).

1. VERSLO PROCESŲ TEISINGUMO TIKRINIMAS

Verslo procesų analizei ir teisingumo tikrinimui gali būti naudojamos tos pačios sąvokos kaip ir programinių sistemų kūrimo procese ar dirbtinio intelekto sistemose. Dažniausiai naudojamos tokios sąvokos, kaip verifikavimas ir validavimas. Daugelis autorių (Pranevicius, 2006) naudoja įvairius verifikavimo ir validavimo metodus (1 pav.), tačiau skirtingi autoriai naudoja jų skirtingus apibrėžimus. Darbe bus nagrinėjami tik modeliais pagrįsti metodai.



1 pav. Formalūs verifikavimo ir validavimo metodai

Modelių tikrinimo metodai naudoja tokias sąvokas: verifikavimas tikrina, ar modelis atitinka jo specifikaciją, o validavimas atsako, ar modelis darys tai, ko iš jo reikalauja vartotojas.

Nors verifikuojant procesus tikrinamos įvairios jų savybės, šiame darbe bus apsiribojama tik pasiekiamumo analizės problemomis: aklaviečių ir neefektyvių ciklų apieška.

Aklavietė – tai sistemos būseną, į kurią galima patekti, tačiau iš jos niekada nepereinama į jokią kitą sistemos būseną.

Uždaras ciklas – tai sistemos būsenų aibė, į kurią patekus neegzistuoja perėjimas į jokiais kitas sistemos būsenas. Kitaip sakant, jei sistema patenka į šias būsenas, ji patenka į amžiną ciklą.

2. VERSLO PROCESŲ MODELIAVIMAS

Dauguma organizacijų aprašo verslo procesus, naudodamos tam nepritaikytus įrankius – tekstinį redaktorių Microsoft Word, laisvos formos diagramų redaktorių Microsoft Visio, prezentacijų redaktorių Microsoft PowerPoint ar atitinkamas programas iš Open Office

programinės įrangos paketo. Tokios veiklos rezultatas yra gana detalus verslo procesų aprašymas. Tačiau kuriant ir diegiant būsimus procesus, tokius aprašymus yra labai sudėtinga palaikyti, kadangi informacija yra netiksli, nestruktūrizuota, nėra semantinių informacijos ryšių.

Formalūs procesų modeliavimo metodai, pavyzdžiui, valdymo srautų, Ganto, PERT diagramos ir kitos atsirado 20-ojo amžiaus pradžioje. Šiandien daugiausia naudojami šiuolaikiniai metodai- tai unifikauta modeliavimo kalba (Unified Modeling Language -UML) ir verslo procesų modeliavimo notacija (Business process modeling notation -BPMN).

Šiame darbe siūloma praplėsti verslo procesų modeliavimo metodus panaudojant procesų modeliavimui IR/ARBA grafus, kurie plačiai naudojami dirbtinio intelekto sistemose.

Verslo procesų modeliavimui formaliai aprašysime "IR/ARBA" grafą.

Apibrėžimas: Verslo proceso logika yra užrašoma "IR/ARBA" tipo grafu $G=(X,W)$:

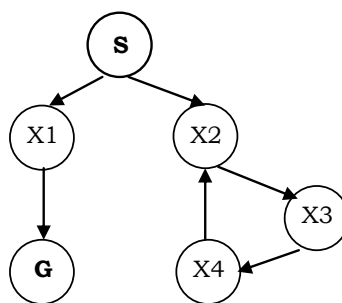
- $X = \{x_1, x_2, \dots, x_n\}$ tai viršūnių rinkinys, kuris išreiškia verslo procesų užduotis. Viršūnės yra IR/ARBA tipo;
- $W \subseteq X \times X$ tai lankų aibė, kuri vaizduoja užduočių srautus;
- Rinkinys $X^s \subseteq X$ yra baigtinis pradžios užduočių rinkinys;
- Rinkinys $X^g \subseteq X$ yra baigtinis tikslo užduočių rinkinys;
- Kelias yra lankų seka $p = x_1, x_2, \dots, x_{k-1}, x_k$ ir visi lankai yra skirtingi $i = 1, 2, \dots, k-1, k$. Kelias yra ciklinis jei $x_i = x_j$;
- Viršūnės $x \in X$ aplinka yra gretimų viršūnių rinkinys žymimas $E(x) = \{u \in X : \{x, u\} \in W\}$. Dvi viršūnės $u, v \in X$ yra gretimos, jei jos sujungtos lanku. Viršūnių išėjimai ir įėjimai yra žymimi atitinkamai $E^+(x)$ ir $E^-(x)$ Bendra viršūnės įėjimų ir išėjimų aibė yra užrašoma aibe $E(x) = E^+(x) \cup E^-(x)$;
- Viršūnės laipsnis yra apibrėžiamas įeinančių ir išeinančių lankų skaičiumi ir žymimas kaip $\deg(x) = |E(x)|$. Įėjimų laipsnis yra žymimas $\deg^-(x)$, o išėjimų laipsnis žymimas $\deg^+(x)$. Laipsnis aprašo $\deg(x) = \deg^-(x) + \deg^+(x)$ įeinančius ir išeinančius lankus iš/į viršūnes;
- Viršūnė $x \in X$ su $\deg^-(x) = 0, \deg^+(x) > 0$ yra vadinama pradžios arba pradine viršūne;
- Analogiškai, tikslo viršūnė yra apibrėžiama kaip $\deg^+(x) = 0, \deg^-(x) > 0$;
- Kitos viršūnės $x \in X$ su laipsniais $\deg(x) = \deg^-(x) + \deg^+(x)$ yra vadinamos vidinėmis grafo viršūnėmis.

Grafe nagrinėjamos pasiekiamumo problemos. Tai aklavietės ir neefektyvūs ciklai. Jie yra apibrėžiami taip:

- Aklavietė - yra vidinė viršūnė $x \in X$, kurios $\deg^+(x) = 0, \deg^-(x) > 0$ ir nepriklauso tikslo viršūnių rinkiniui;
- Neefektyvus ciklas - tai ciklinė viršūnių seka, dėl kurios atsiranda nepasiekiamų viršūnių.

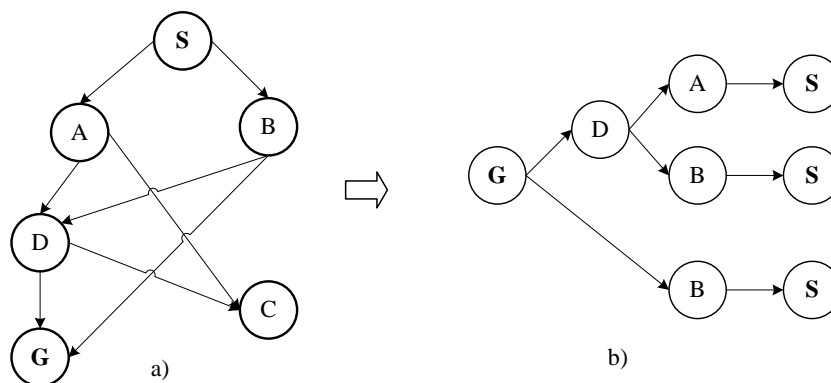
Pasiekiamumo problemas spręsimė naudodami modifikuotą paieškos algoritmą grafuose.

Sprendžiant pasiekiamumo problemą reikia išanalizuoti visus kelius kurie prasideda pradžios taške S ir baigiasi tikslo taške G. Visi keliai p_j nuo pradžios viršūnės S iki tikslo viršūnės G gali būti išskaidomi į $S \xrightarrow{p_{sj}} X_j \xrightarrow{p_{jg}} G, j=1,2,3$. Pasiekiamumo problemos atsiranda, kai egzistuoja aklavietės ar neefektyvūs ciklai. Pavyzdžiui, (2 pav.) x_2, x_3, x_4 viršūnių neįmanoma pasiekti tikslo viršūnės G, jeigu pradėsime iš starto viršūnės S. Tai veda prie pasiekiamumo savybių pažeidimo. Pavaizduotame grafe viršūnes x_2, x_3, x_4 sudaro ciklą.



2 pav. Pasiekiamumo problemos pavyzdys

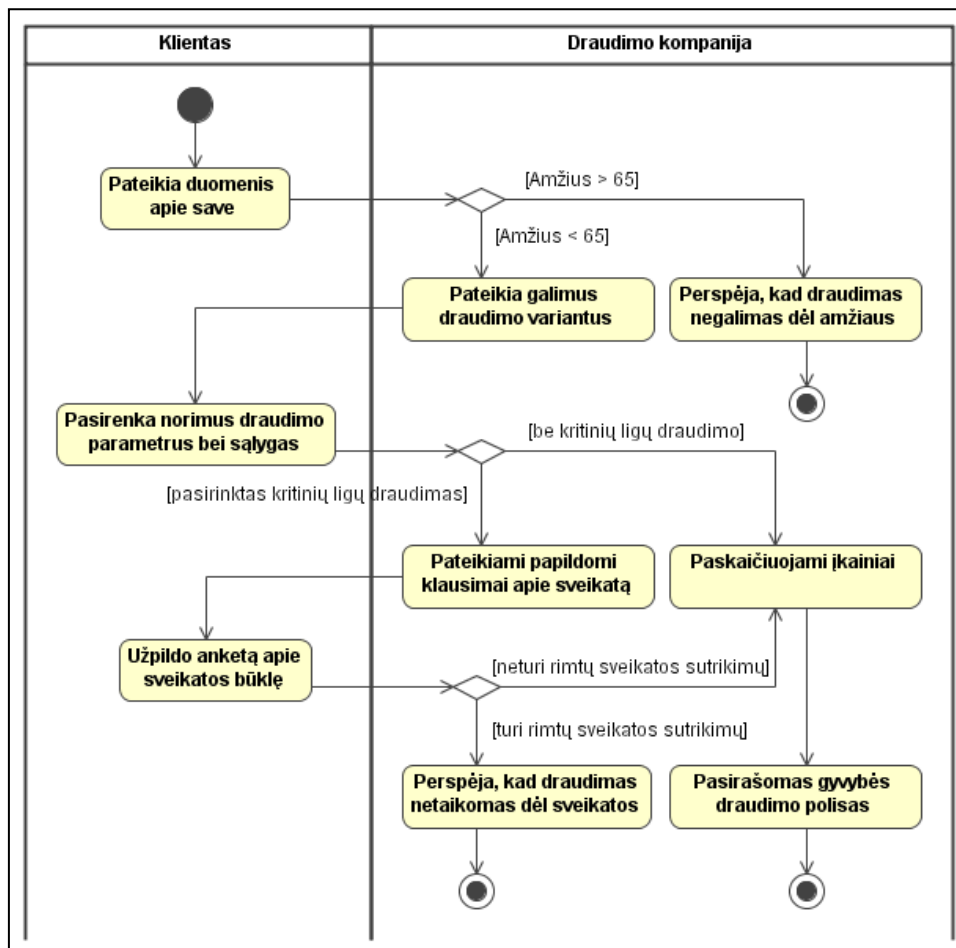
Šiame darbe verslo procesų verifikavimui bus panaudotas modifikuotas paieškos gilyn algoritmas orientuotiems grafams. Algoritmas pritaiko rekursinę paiešką gilyn kiekvienai viršūnei ir randa jos visus kelius tarp pradžios ir galinės viršūnių. Šis paieškos algoritmas gali analizuoti ir tokius grafus, kuriuose yra ciklą. Šio algoritmo principas pavaizduotas paveiksle (3 pav.). Šiame paveiksle viršūnė C yra nepasiekiamama kelyje tarp pradinės S ir galinės G viršūnių, todėl paveikslo b) dalyje jos nėra.



3 pav. Modifikuoto algoritmo veikimo principas

3. VERSLO PROCESO PAVYZDYS

Verifikavimo algoritmui pademonstruoti panagrinėkime pavyzdį, kuriame vaizduojamas sveikatos draudimo procesas (Šilingas, 2013).

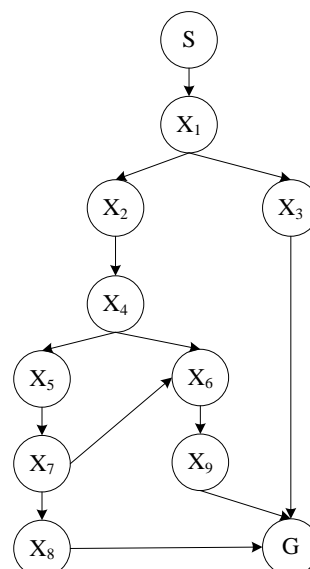


4 pav. Veiklos diagrama - gyvybės draudimo procedūros aprašymas

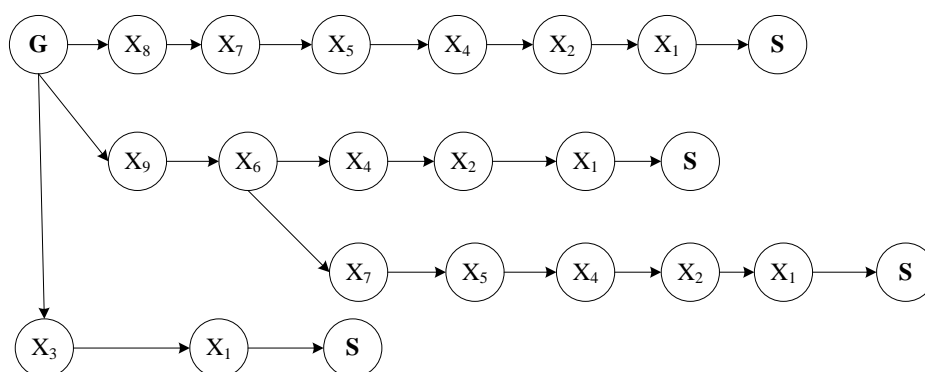
Analizuojant šį procesą, pirmiausia gyvybės draudimo procedūra pavaizduojama IR/ARBA grafu (5 pav.). Išanalizavus grafą, nustatyta, kad gyvybės draudimo procesas užrašytas teisingai, nes visos viršūnės pasiekiamos iš pradinės startinės ir veda į galinę būseną (6 pav.)

Pagal veiklos diagramą įvesti tokie pažymėjimai:

- X1 – pateikia duomenis apie save;
- X2 – pateikia galimus draudimo variantus;
- X3 – perspėja, kad draudimas negalimas dėl amžiaus;
- X4 – pasirenka norimus draudimo parametrus bei sąlygas;
- X5 – pateikiami papildomi klausimai apie sveikatą;
- X6 – paskaičiuojami įkainiai;
- X7 – užpildo anketą apie sveikatos būklę;
- X8 – perspėja, kad draudimas netaikomas dėl sveikatos;
- X9 – pasirašomas gyvybės draudimo polisas.



5 pav. Gyvybės draudimo procedūros grafas



6 pav. Gyvybės draudimo grafo analizės rezultatai

IŠVADOS

Šiame darbe pasiūlyta verslo procesų modeliavimui panaudoti grafus, o jų teisingumą tikrinti panaudojant paieškos algoritmus grafuose. Šio metodo privalumas toks, kad jis leidžia grafiškai aprašyti nesudėtingą procesų logiką, o tokia grafų forma gerai suprantama tiek verslo atstovams, tiek ir informacinių technologijų specialistams. Darbe pateiktas modifikuotas algoritmas leidžia aptikti aklavietes ir neefektyvius ciklus.

LITERATŪRA

1. Business Process Model and Notation (BPMN) version 2.0 (2011). [žiūrėta 2013 m. birželio 10 d.]. Prieiga per internetą: <<http://www.omg.org/spec/BPMN/2.0/PDF>>
2. Gudas, S.; Lopata A. (2005). Darbų sekų modeliais grindžiamas veiklos žinių surinkimo būdas, *Informacijos mokslai*, t. 34, p. 277-278.
3. Jakštonytė G. Gyventojų pajamų mokesčio deklaracijos grafinis modelis (2006), *Ekonomika ir vadyba: aktualijos ir perspektyvos*, 2 (7), p. 77-84.
4. Pranevičius H.; Misevičienė R. (2012). Verification of business process workflows, *Technological and economic development of economy*. London : Taylor & Francis ; Vilnius : Technik, Vol. 18, no. 4, p. 623-635.
5. Pranevičius, H., Misevičienė, R. (2006). Verification of piece-linear aggregate specifications. Monograph. Kaunas: Technologija, 268 p.
6. Ross, R. G. (2005). How Rules and Processes Relate. Part 2. Business Processes. *In Business Rules Journal*, Vol. 6, No. 11, [žiūrėta 2013 m. birželio 10 d.]. Prieiga per

- internetą: <<http://www.brcommunity.com/b256.php>>.
7. Weske M. (2007). *Business Process Management: Concepts, Languages, Architectures*, Springer. [žiūrėta 2013 m. birželio 10 d.]. Prieiga per internetą: <http://books.google.lt/books?id=QMyu_B1KTZIC&redir_esc=y>
 8. Vadybos pokyčių konsultavimas. [žiūrėta 2013 m. birželio 10 d.]. Prieiga per internetą: <www.konsultavimas.lt>
 9. Šilingas D. ir Milevičienė E. (2013). Verslo procesų modelių tobulinimas: anti-pavyzdžių pertvarkymas pritaikant gerąsias praktikas. *No Magic Europe*. [žiūrėta 2013 m. birželio 10 d.]. Prieiga per internetą: <<http://www.vpvp.lt/straipsniai>>
 10. Verslo procesų valdymas. [žiūrėta 2013 m. birželio 10 d.]. Prieiga per internetą : <<http://estrategija.lt/23ProcesuZemelapis/ProcesuZemelapis.htm>>