

KAUNO TECHNOLOGIJOS UNIVERSITETAS
VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

DARIUS AŠERIŠKIS

**PROGRAMŲ SISTEMŲ ŽAIDYBINIMO ELEMENTŲ
MODELIAVIMAS IR VERTINIMAS**

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (07T)

2017, Kaunas

Disertacija rengta 2013–2017 metais Kauno technologijos universiteto Informatikos fakultete Programų sistemų katedroje.

Mokslinis vadovas:

Prof. dr. Robertas DAMAŠEVIČIUS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, 07T)

Redagavo: Bronius Broniauskas (Publishing Office “Technologija”)

Informatikos inžinerijos mokslo krypties daktaro disertacijos gynimo taryba:

Prof. dr. Rimvydas SIMUTIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, 07T);

Prof. dr. Rimantas GATAUTIS (Kauno technologijos universitetas, socialiniai mokslai, vadyba, 03S);

Prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, 07T);

Prof. dr. Tomas KRILAVIČIUS (Vytauto Didžiojo universitetas, technologijos mokslai, informatikos inžinerija, 07T);

Doc. dr. Ricardo QUEIRÓS (Aukštesniojo medijos menų ir dizaino mokykla, Porto politechnikos institutas, technologijos mokslai, informatikos inžinerija, 07T).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2017 m. spalio 27 d. 13 val. Kauno technologijos universitete, Disertacijų gynimo salėje (K. Donelaičio g. 73, 403 a., Kaunas).

Adresas: K. Donelaičio g. 73-403, 44249 Kaunas, Lietuva.

Tel. (370) 37 300 042; faks. (370) 37 324 144; el. paštas doktorantura@ktu.lt.

Disertacijos santrauka išsiųsta 2017 m. rugsėjo 27 d.

Su disertacija galima susipažinti internetinėje svetainėje <http://ktu.edu>, Kauno technologijos universiteto (K. Donelaičio g. 20, 44239 Kaunas) ir Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, LT-10223 Vilnius) bibliotekose.

TERMINAI IR SANTRUMPOS

ARCS

Angl. *Machinations*

dėmesys, aktualumas, pasitikėjimas ir pasitenkinimas.

teorinė struktūra, kur žaidimai apibrėžiami kaip dinaminės sistemos ir daugiausia dėmesio skiriama jų grįžtamojo ryšio ciklams.

Mechanika, dinamika, estetika (angl. Mechanics-Dynamics-Aesthetics (MDA) Metakalba

mechanika, dinamika, estetika: žaidimų projektavimo srityje – priemonė, naudojama žaidimams analizuoti.

Mažumos žaidimas (angl. Minority game, MG)

tam tikros formos kalba arba terminų aibė, skirta kitai kalbai apibūdinti ar analizuoti.

paprastas modelis, kuris aprašo kolektyvinę veikėjų elgseną idealioje situacijoje, kur jie turi konkuruoti dėl baigtinio išteklių kiekio.

Struktūra (angl. Pattern)

reguliari ir suprantama forma arba seka, kuria galima pastebėti, kai kas nors atsitinka ar būna daroma.

Petri tinklai (angl. Petri Nets)

viena iš kelių matematinio modeliavimo kalbų, skirtų paskirstytosioms sistemoms aprašyti.

Sistemų pritaikymo galimybių skalė (angl. System Usability Scale, SUS)

elementų Likerto skalė, leidžianti susidaryti bendrą supratimą apie subjektyvų sistemų naudojimo galimybių įvertinimą.

Sistemų modeliavimo kalba (angl. Systems Modeling Language, SysML)

modeliavimo kalba, skirta sistemų inžinerijos programoms modeliuoti.

Unifikuota modeliavimo kalba (Unified modeling language, UML)

bendrosios paskirties plėtros ir modeliavimo kalba programinės įrangos sistemų sričiai, skirta užtikrinti standartinį būdą vizualizuoti sistemos sandarą.

1. ĮVADAS

1.1. Motyvacija

Pastaruosiu metu žaidybinimas tampa vis populiariesnis įmonių informacijos bei el. komercijos sistemų plėtros srityje. Žaidybinimas apibūdinamas kaip žaidimo elementų pritaikymas aplinkai, kuri nėra susijusi su žaidimais [1], pavyzdžiui, rinkodaros, darbuotojų veiklos rezultatų įvertinimo, darbuotojų mokymo, inovacijų valdymo srityse.

„Pew“ tyrimų centro apklausa, 53 % apklaustųjų teigė, kad 2020 m. žaidybinimas jau bus plačiai paplitęs [2]. Gerai žinomo tyrimo autorius Gartneris teigė, kad iki 2015 m. daugiau nei 50 % visų organizacijų sužaidybins inovacijų diegimo procesus, siekdamas efektyviau juos valdyti [3]. Modeliai, pagrįsti žaidimo teorija, šiuo metu plačiai taikomi skirtinguose kontekstuose ir naudojami įvairiausiose sferose kaip problemų sprendimo įrankis.

Esami žaidybinimo vertinimai paprastai skiria pagrindinį dėmesį vartotojų apklausoms ir kitokiems kokybinio vertinimo metodams. Vis dar trūksta modeliavimo metodų ir priemonių, kurie galėtų padėti įgyvendinti bei plėtoti rimtų sistemų žaidybinimą [4], [5].

Šioje disertacijoje siekiama pristatyti modeliavimo ir imitavimo metodus, kurie galėtų tapti tarpine grandimi tarp formalaus ir kiekybinio

sužaidybintų sistemų modeliavimo, taip pat padėtų nagrinėti ir įvertinti žaidimų taisykles ir procesus.

1.2. Tyrimo objektas ir taikymo sritis

Tyrimo objektas yra sužaidybtos programinės įrangos sistemų modeliavimo, analizės ir vertinimo metodai bei priemonės. Šie metodai ir priemonės yra būtini, jei norime pritaikyti veiksmingesnius žaidimų modelius, elementus ir mechaniką bei diegti juos įvairiose sistemose, ir kartu suprasti, kokią įtaką tokios sistemos turės vartotojų elgsenai.

Tyrimas apima:

- Žaidimų modeliavimo, analizės, prototipų kūrimo metodus.
- Sužaidybintų sistemų formalų matematinį modeliavimą.
- Vartotojų elgsenos modeliavimą bei modelių elgsenos analizę.
- Sužaidybintų sistemų analizę ir vertinimą.

1.3. Problemos formulavimas ir tyrimo klausimai

Pagrindinė šiame darbe analizuojama problema – kiekybinės analizės metodų trūkumas bei nepakankamas sužaidybtoms sistemoms būdingų struktūrų, elementų ir mechanikos supratimas. Šiuo metu nėra nė vieno proceso, kuris leistų žaidimų projektuotojams pereiti nuo žaidybinimo idėjos iki galutinio sužaidybtos sistemos įgyvendinimo.

Disertacijoje bus atsakyta į šiuos tyrimo klausimus:

- Kokios yra pačios naujausios sužaidybintų sistemų modeliavimo tendencijos?
- Kaip turėtumėme įvertinti sužaidybtas sistemas?
- Kaip turėtų vykti abstraktus sužaidybintų sistemų bei jų elementų modeliavimas?
- Kaip numatyti sužaidybintų sistemų vartotojų elgseną?

1.4. Tikslai ir užduotys

Šio tyrimo tikslas – pasiūlyti žaidybinimo priemonių ir metodų, kurie leistų modeliuoti, analizuoti, išbandyti bei kurti sužaidybtas sistemas, išskirti pavienes struktūras, perprasti sužaidybintų struktūrų poveikį sužaidybtoms sistemoms ir įvertinti jų įtaką vartotojų elgsenai.

Nurodytam tikslui pasiekti buvo suformuluotos šios užduotys:

1. atlikti sužaidybintų sistemų statinę ir dinaminę analizės, siekiant nustatyti metodus, kuriais galima įvertinti sistemos žaidybinumą;
2. išnagrinėti sužaidybintų sistemų struktūras ir nustatyti jų bendrus bruožus, pasiūlyti sužaidybintų sistemų modeliavimo metodą;
3. išnagrinėti žinomus sprendimus, susijusius su sužaidybintų sistemų veiksmingumo ir pritaikymo galimybių įvertinimu bei pasiūlyti metodą, leidžiantį analizuoti ir matematiškai sumodeliuoti

sužaidybinių sistemų įtaką vartotojų elgsenai, tiesiogiai susijusiai su šiomis sistemomis.

1.5. Ginami teiginiai

Ginami šie teiginiai:

1. siūlomi sužaidybinių sistemų vertinimo metodai leidžia atlikti jų kiekybinį ir kokybinį įvertinimą;
2. siūlomas sužaidybinių sistemų vaizdinio modeliavimo būdas leidžia konstruoti sužaidybinių sistemų modelius, išskirti šioms sistemoms būdingas struktūras, modeliuoti vartotojų elgseną ir ją analizuoti, lyginti sužaidybinis sistemas ir kurti sužaidybinių programinę įrangą;
3. UAREI modelių imitavimo funkcija GMOD programiniame įrankyje padeda atkurti panašų kitų įrankių elgesį, taip pat nuspėti, kaip įgyvendinamas sužaidybinės sistemos prototipas veiks realioje aplinkoje. Sužaidybinių sprendimų veiksmingumą galima sumodeliuoti ištyrus vartotojų elgseną, o pastarajam tikslui galima naudoti HEXAD klausimyno pagrindu atliekamą įvertinimą, kuris paprastai taikomas įprastinių žaidimų vartotojams.

1.6. Pagrindinis indėlis ir naujumas

Pagrindinis indėlis į tiriamą sritį:

- sukurtas formalus, abstraktus sužaidybinių sistemų modeliavimo metodas. Šis metodas leidžia modeliuoti, analizuoti ir vertinti sužaidybinis sistemas. Metodas turi panašias arba geresnes savybes lyginant su kitais taikomais metodais;
- sukurtas naujas sužaidybinių sistemų įdomumo vertinimo metodas yra parentas žaidėjų pergalių pasiskirstymo lyginimu su normaliuoju skirstiniu. Metodas išbandytas analizuojant mažumos žaidimo variacijas;
- sukurtas naujas metodas skirtas sužaidybinių sistemų stiprinamųjų metodų vertinimui pagal psichologinius žaidėjų tipus. Pasiūlytas metodas išbandytas *Oil Trader* eksperimentu;
- pasiūlyti metodai sužaidybinių sistemų grafinei sąsajai vertinti, taikant WCAG ir SUS, patikrinti *Trogon* projektų valdymo sistema;
- pasiūlyti metodai ir modeliavimo metodas leidžia žaidimų projektuotojams ir mokslininkams plėtoti sužaidybinių sistemų modelius, modeliuoti sužaidybinių sistemų veiklą, gerinti turimus modelius, siekiant norimų rezultatų bei modelių pagrindu plėtoti sistemas. Tai pagreitina sužaidybinių sistemų vystymą.

Metodo naujumas:

- UAREI yra naujas formalaus modeliavimo metodas, skirtas sužaijdybintoms sistemoms, pasižyminčioms vizualizavimo ir imitavimo galimybėmis, modeliuoti;
- sukurtas metodas leidžia skirtingais būdais imituoti sužaijdybintų sistemų veikimą. Imitavimui UAREI pasitelkia parinkimo funkcijas, veikėjų pagrindo modeliavimą ir mažumos žaidimo variklį;
- sužaijdybintų sistemų imitavimas leidžia vertinti sužaijdybintų sistemų veiklą pagal žaidėjų motyvaciją, pasiektus rezultatus ir žaidimo įdomumą;
- sukurti sužaijdybintų sistemų vertinimo metodai yra paremti panaudojamumu, patrauklumu (spalvų kontrastu) ir žaidėjų motyvacijos vertinimu pagal žaidėjų tipus.

1.7. Praktinė reikšmė

Praktinė tyrimo reikšmė:

- siūlomas formalus abstraktus modelis leidžia mokslininkams atlikti formalų sužaijdybintų sistemų modeliavimą. Tos pačios formalios notacijos naudojimas leidžia lengvai išskirti bendras sužaijdybintoms sistemoms būdingas struktūras ir įtraukti šių struktūrų sudėtį į naujas sistemas;
- sužaijdybintų sistemų modeliavimas suteikia žaidimų projektuotojams ir mokslininkams vertingų išvalgų apie tai, kokį poveikį sužaijdybintos sistemos daro vartotojų elgsenai. Modeliavimas užtikrina greitesnį grįžtamąjį ryšį ir padeda prognozuoti elgesį;
- programinės įrangos kūrimas UAREI metodu – tai trumpiausias kelias nuo sužaijdybintos sistemos modelio iki sistemos prototipo;
- sužaijdybintų sistemų analizė UAREI modeliavimo metodu leidžia geriau suprasti, kaip sužaijdybintų sistemų struktūros veikia galutinio vartotojo elgesį, kaip šios struktūros sąveikauja tarpusavyje ir kaip pasiekti geriausių sužaijdybintos sistemos veiklos rezultatų prieš diegiant paruoštas naudoti sužaijdybintas programinės įrangos sistemas.

1.8. Mokslinis pripažinimas

Tyrimo rezultatai buvo pristatyti dviejose tarptautinėse konferencijose, du straipsniai buvo paskelbti moksliniuose žurnaluose, įtrauktuose į *Web of Science* sąrašą. Dar vienas straipsnis turi būti išleistas moksliniame žurnale, įtraukta į *Web of Science* sąrašą. Vienas straipsnis paskelbtas recenzuojamame žurnale. Trys susiję mokslo darbai buvo pristatyti konferencijose Lietuvoje. Du mokslo darbai pristatyti konferencijose Ispanijoje ir Prancūzijoje. Visą leidinių sąrašą galima rasti skyriuje

„DARIAUS AŠERIŠKIO PUBLIKUOTŲ DARBŲ SĄRAŠAS DISERTACIJOS TEMA“. Konferencijų sąrašas:

- The 22th International Master and PHD Students Conference “Information Society and University Studies”, 2017 April 28, Kaunas;
- The Seventh International Conference on Advances in Computer-Human Interactions ACHI 2014 March 23 - 27, Barcelona, Spain;
- The Sixth International Conference on Intelligent Human Computer Interaction IHCI 2014 December 8-10, Envy, France;
- The 18th International Master and PHD Students Conference “Information Society and University Studies”, 2013 April 25, Kaunas;
- The 17th Master and PHD Students Conference “Information Society and University Studies”, 2012, Kaunas.

2. ŽAIDYBINIMO ANALIZĖ

2.1. Žaidybinimo psichologiniai bei socialiniai aspektai

Žaidybinimą galima paaiškinti *Fogg* elgsenos modeliu (*Fogg Behavior Model*, *FBM*) [6], kuriame teigiama, kad tam tikras elgsenys atsiranda tuomet, kai turima tiek motyvacija, tiek gebėjimas atlikti atitinkamą veiksmą. Išsamiau psichologinį žaidybinimo pagrindą nagrinėjo Wu [7], kuris tyrė, kaip ir kiek žaidybinimas gali skatinti asmens veiksmus, ir Gnaukas ir kt. [8], kurie tyrė išorinę bei vidinę motyvaciją ir analizavo jos santykį su išorinėmis lengvatomis bei paskatomis.

Daugelis sužaidybtųjų sistemų skatina vartotojo dalyvavimą, naudojantis virtualiomis paskatomis, pavyzdžiui, taškais, ženkleliais, reitingais [9], pažangos juostelėmis, veiklos grafikais ar avatarais [10]. Šios paskatos pateikia žaidėjo paaukotą laiką ir pastangas tokiu pavidalu, kuris gali būti išreikštas kiekybiškai, kitiems žaidėjams suprantama forma ir palyginta su jų rezultatais. Taip šios priemonės rodo žaidėjo statusą ir pažangą žaidime, taip pat skatina jį žaisti toliau.

Kita vertus, jei vienam iš žaidėjų sekasi žaisti žymiai geriau negu kitiems, ir jis, žinoma, nuolat laimi, tiek šis pastovus nugalėtojas, tiek kiti žaidėjai ir žiūrovai gali pradėti jausti šiokių tokių nuobodulį. Nuobodulys skatina siekti kitokių tikslų, nepriklausomų nuo žaidimo [11], ir taip sumažina likusių žaidėjų skaičių.

Jei norima to išvengti, motyvacijos skatinimo modelis turėtų būti kruopščiai pritaikytas įgūdžių skirtumams panaikinti tarp žaidėjų ir skatinti juos tęsti žaidimą. Jei motyvacijos skatinimo modelis yra tinkamai subalansuotas, jis gali sukelti žaidėjams itin motyvuojančią emocinio srauto būseną [12].

Remdamiesi srauto teorija [12], Chanelis ir kt. [13] apibrėžė tris skirtingas emocines būsenas: nuobodulį (neigiamą ir ramų), susidomėjimą

(teigiama ir jaudinanti) ir nerimą (neigiamą ir jaudinanti). Malonumas, pagal srauto teoriją, pasireiškia ties nuobodulio ir nerimo riba, kai iššūkiai būna tiksliai suderinti su asmens gebėjimu dalyvauti žaidime [12]. Susidomėjimas ir įsitraukimas iki šiol buvo dažniausiai apibrėžiami per tokias pažintines ir psichologines būsenas kaip dalyvavimas, buvimas ir susijaudinimas, kurios prisideda prie susidomėjimo [14].

2.2. Sužaidybintų sistemų analizės metodų apžvalga

Dauguma žaidimų turi tam tikrų bendrų aspektų. Pagrindinė idėja, kuria remiasi struktūrų samprata – tai dažniausiai pasikartojančių problemų struktūrinių sprendimų apibrėžimas ir formalizavimas. Struktūrą paprastai sudaro pavadinimas, apibrėžimas, bendrasis aprašymas, galimų struktūros naudojimo būdų aprašymas, struktūros naudojimo pasekmių aprašymas ir santykiai su kitais modeliais [15]. Per projektavimo struktūrų sąvoką [15], kuri sėkmingai naudojama objektinio projektavimo ir programinės įrangos inžinerijos srityse ir kol kas pateisino visus lūkesčius, siekiama išreikšti šiuos sprendimus lengvai suprantama forma.

Kreimeieris [16] pasiūlė naudoti struktūras žaidimų projektavimo srityje kaip būdą įteisinti ir reglamentuoti žinias apie žaidimų sandarą. Bjorkas ir Holopainen [17] pasiūlė apibūdinti žaidimo strategijos struktūras kaip dažnai pasikartojančių strateginių žaidimo dalių pusiau formalius, tarpusavyje susijusius aprašymus. Žaidimams būdingos struktūros jungia tiek bendras sandaros problemas, tiek jų sprendimus, todėl, siekdami sukurti geresnę strategiją, žaidimų kūrėjai paprastai sujungia kelias struktūras į vieną [18].

Pats žaidybinimo procesas gali būti apibrėžtas 4-iomis teorijomis [19]: žaidimo sandaros ypatumų (angl. Game Design Features – GDF), pagrindinių mokomojo žaidimo savybių (angl. Key Characteristics of a Learning Game – CLG), dėmesio, aktualumo, pasitikėjimo ir pasitenkinimo (angl. Attention, relevance, confidence, and satisfaction – ARCS) bei žaidimo mechanikos, dinamikos ir estetikos (angl. Game mechanics, dynamics and aesthetics – MDA).

Sunku lyginti GDF su kitomis teorijomis (el. mokymosi prasme), nes ji buvo skirta daugiausiai linksmiems kompiuteriniams žaidimams kurti. Tačiau vis dėlto ji glaudžiai koreliuoja su CLG ir suteikia el. mokymuisi daugiau įdomumo, motyvacijos ir patrauklių iššūkių, kadangi sukuria geresnę patirtį, panašią į įprastinio žaidimo strategiją. ARCS taikoma taip pat kaip ir CLG, ir gauna tiesioginės naudos nuo pastarajai priskiriamų savybių – didesnio susidomėjimo, dėmesingumo, pasitikėjimo, o galiausiai ir didesnio pasitenkinimo mokymosi procese, kuris atsiranda dėl sėkmingai pasiektų tikslų [20]. Tačiau klasikinė žaidimų MDA teorija taikoma tik keletui CLG aspektų, tokių kaip domėjimasis arba iššūkiai, susiję su dinamika ir sudėtingumo lygiu [21].

UML yra de facto standartinė modeliavimo kalba, naudojama daugelyje sričių. Tenzeris [22] teigia, kad UML modeliavimo įrankiai gali būti

naudojami žaidimams kurti, ir siūlo sistemą, kuri leistų projektuoti žaidimus naudojantis UML. SysML yra bendrosios paskirties modeliavimo kalba, skirta sistemų inžinerijos programoms ir padedanti detalizuoti, analizuoti, projektuoti bei patvirtinti įvairias sistemas.

Ryškesni žaidimų aprašymo kalbų pavyzdžiai, priklausantys nuo sferos specifikos – tai GaML [23] ir ATTAC-L. GaML yra formalizuota kalba, skirta sužaidybintiems sprendimams detalizuoti ir automatiškai generuoti. Tai išlaisvina IT ekspertą nuo būtinumo pačiam plėtoti sužaidybintus sprendimus. ATTAC-L yra nuo sferos specifikos priklausanti kalba, kuri leidžia vartotojui nurodyti žaidimo scenarijų XML ir sukurti žaidimą, naudojant kodų generatorių.

Rimtų žaidimų logikos ir struktūros modeliavimo kalba (angl. Serious Game Logic and Structure Modeling Language – GLiSMo) [24], kurią pasiūlė Thillainathanas, numato modeliavimo sistemą, susidedančią iš dviejų modelių: struktūros ir logikos. Siūloma kalba buvo skirta pedagogams, kurie nedirba technikos srityje, tačiau turi galimybę kurti rimtus žaidimus.

Kitas požiūris į sužaidybintų procesų modeliavimą remiasi formaliais (arba matematiniais) modeliais. Bista ir kt. [25] pasiūlė pirmąjį formalų sužaidybinio sistemos modelį. Chanas ir kt. [26] siūlo panašų požiūrį į socialinių žaidimų modeliavimą, kuris, be to, leidžia patikrinti sukurtą modelį. Oliveira ir kt. [27] modeliuoja žaidimus, naudodamiesi Petri tinklais. Šio metodo trūkumas yra nepakankamos galimybės pritaikyti jį prie atitinkamos sferos specifikos; būtent todėl jis iki šiol nėra taikomas žaidimų projektuotojų.

Abstrakčiau žaidimo elementus gali padėti aprašyti žaidybinimo modeliavimo kalba (angl. Gamification Modeling Language (GaML) [23], kuri buvo sukurta XML pagrindu. Ji suteikia mechanizmą, kuriuo galima apibrėžti žaidybinimo sąvokas tiksliai bei taip, kad jos tiktų apsigėtimui su žaidimo mechanika. Galiausiai žaidimo taisyklės sujungia žaidimo elementus į žaidimo sluoksnį. Tokias žaidimo taisykles galima sumodeliuoti pasitelkus *Machinations* vizualinio modeliavimo notaciją, kuri buvo sukurta Petri tinklų pagrindu.

Trečiąją sužaidybintų sistemų modeliavimo metodų kategoriją sudaro vaizdinės kalbos, skirtos greitai kurti prototipus žaidybinimo sferoje. Žinomiausi pavyzdžiai yra *Sketch-It-Up*, *Ludocore* ir *Machinations*. *Sketch-It-Up* yra įrankis, leidžiantis kurti galimų žaidimų eskizus.

2.3. Santrauka

Žaidybinimas yra metodika, naudojama siekiant patobulinti programinės įrangos sistemas, pritaikant joms žaidimo mechaniką ir žaidimo elementus. Pagrindinis motyvas, kuriuo remiasi žaidybinimas – tai siekis pakeisti vartotojų elgseną. Žaidybinimas naudojamas skirtingiems tikslams. Buvo apibrėžti psichologiniai modeliai ir teorijos, paaiškinantys, kodėl ir kaip žaidybinimas keičia vartotojų elgseną pageidaujama linkme. Žaidybinimo poveikis yra linkęs nykti ir reikalauja paskatinimo modelių, kurie sulėtintų šį

irimo procesą. Pagrindiniai emociniai veiksniai ir būsenos, kuriuos reikia įtraukti į paskatinimo modelius, yra šie: nuobodulys, nuovargis, nusivylimas, pasitenkinimas, grįžtamasis ryšys ir suinteresuotumas.

Sužaidybintą sistemą galima modeliuoti įvairiais būdais, pavyzdžiui, naudojantis UML, MDA, GaML, *Machinations*, Petri tinklus ir pan. Šie metodai turi savų privalumų ir trūkumų.

Sužaidybinimo sferoje trūksta į sritį orientuoto ir tikslingo metodo, skirto spręsti šiai sričiai būdingas problemas, kaip antai:

- žaidimų ir sužaidybintų sistemų modelių analizė, struktūrų išskyrimas, struktūrų poveikio įvairioms programoms perpratimas;
- abstrakčių metodų, kurie leistų atlikti sistemos sužaidybinimo modeliavimą bei sumodeliuoti tokios sužaidybinios sistemos veiklą;
- kiekybinio sužaidybintų sprendimų įvertinimo būdų;
- priemonių, kurios leistų modeliuoti, imituoti, analizuoti ir generuoti sužaidybintus sprendimus bei toliau siekti gilesnio sužaidybintų sistemų supratimo;
- būdų modeliuoti sužaidybinias sistemas, atsižvelgiant į psichologiškai pagrįstus vartotojo elgsenos tipus.

3. SUŽAIDYBINTŲ SISTEMŲ DETALIZAVIMAS

3.1. Sužaidybintų sistemų analizės metodologija

Septynios skirtingos sužaidybinios programos (*Emo-bin*, *Meeco*, *Teamfeed*, *CAPTCHINO*, *Taskville*, *Power House*, *Trogon*) buvo atrinktos žaidybiniams struktūroms atpažinti ir analizuoti. Naudodamiesi žaidimų modeliavimo įrankiu *Machinations*, kiekvienai programai sukūrėme dviejų tipų modelius: 1) paprastas modelis – aukščiausias abstrakcijos lygmuo sistemoje, kuris parodo pagrindines sistemos sąvokas; 2) išplėstinis modelis, kuris yra sudarytas iš dviejų dalių: a) statinis modelis, turintis kiek įmanoma daugiau išsamios informacijos apie sistemą, b) dinaminis modelis, kuris atspindi sąveiką tarp žaidėjų. Be to, kiekvienam modeliui buvo sukurtas formalus modelio aprašas ir tekstinis apibūdinimas Modelių tarpusavio palyginimo ir analizės rezultatai leido mums nustatyti bendras sužaidybintų sistemų struktūras.

3.2. Sužaidybintų sistemų struktūros

Siekdami pavaizduoti modelius grafiškai, naudojome *Machinations* vaizdinę žaidimų mechanikos modeliavimo sistemą [28], kuri palengvina vidinio išteklių pasiskirstymo projektavimą, modeliavimą ir išbandymą įvairiais abstrakcijos lygmenimis. Sistemos pagrindą sudaro grafinis žymėjimas, skirtas žaidimų dinamikai fiksuoti. *Machinations* įrankiu sukurtos diagramos yra Petri tinklo atmaina, kuriai buvo suteikta daugiau formalumo, siekiant padaryti jas priimtinesnes žaidimų projektuotojams. *Machinations*

remiasi tokia logika: žaidimo taktiką galiausiai turi lemti išteklių srautas. Išteklių srautai leidžia įsivaizduoti, kaip sistema buvo sukonstruota ir kokios grįžtamojo ryšio struktūros egzistuoja bendroje žaidimo struktūroje.

Taip pat naudojama ir ši, perimta iš UML modelių aprašymo schema [15]: ketinimai, motyvacija, pritaikymo galimybės, sandara, dalyviai, bendradarbiavimas, pasekmės, įgyvendinimas, pavyzdžiai, susiję modeliai ir aptarimas.



1 pav. (a) Begalinis išteklių kiekis ir (b) ribotas išteklių kiekis

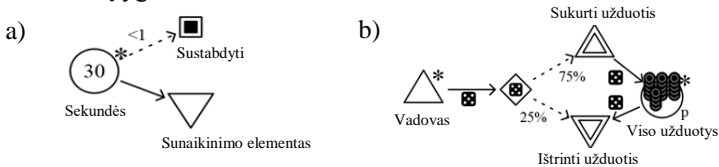
Begalinis išteklių kiekis (žr. 1.a pav. ir 1 lentelę) – šiuo atveju laikoma, kad didžiausio taškų skaičiaus, koks jis bebūtų, surinkti neįmanoma (pavyzdžiui, vartotojo veiksmų skaičiaus neįmanoma nustatyti).

Ribotas išteklių kiekis (žr. pav. 1b ir 1 lentelę) – sistemoje nustatomas apribojimas, t. y. didžiausias taškų skaičius, kurį galima surinkti, ribojamas bet kurioje žaidimo stadijoje. Riba gali būti fizinė ar virtuali. Pavyzdžiui, *Emo-Bin* turimas baigtinis skaičius butelių, apribotas vietinio automato.

1 lentelė. Riboto išteklių kiekio ir begalinio išteklių kiekio struktūrų aprašymas

Savybė	Ribotas išteklių kiekis	Begalinis išteklių kiekis
Ketinimai	Tam tikram ištekliui nustatoma riba	Modeliuojama ekonomika, kur turimi ištekliai neriboti
Motyvacija	Leidžia sumodeliuoti ribotą ekonomiką	Kartais išteklius galima suvokti kaip neribotus. Tai leidžia sumodeliuoti neribotą ekonomiką
Taikymo galimybės	Modeliuojama ekonomika, kur turimų išteklių kiekis ribotas	Modeliuojama ekonomika ar ekonomikos dalis, kur nėra jokių ekonominių apribojimų
Sandara	Naudojama saugykla, turinti automatinį perdavimą	Šaltinio elementas
Dalyviai	Saugykla	Šaltinio elementas
Pasekmės	Ekonominis augimas ribojamas	Leidžiamas neribotas augimas
Pavyzdžiai	<i>Trogon</i> , <i>TaskVille</i> , <i>Emo-bin</i> , <i>Captchino</i>	<i>Teamfeed</i> , <i>Meeco</i> , <i>PowerHouse</i>
Susijusios struktūros	Visos	Visos

Šalia šių dviejų savybių galime pridėti papildomą apribojimą ar labiau realistines sąlygas:



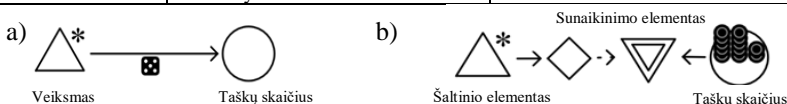
2 pav. (a) Termino (b) ir dinaminės ribos struktūros

Terminas (žr. 2.a pav. ir 2 lentelę) įtraukia į sistemą laiko apribojimą. Jei toks apribojimas taikomas ištekliams, kurių kiekis neribotas, jie tampa apriboti laiko.

Dinaminė riba (žr. pav. 2.b ir 2 lentelę) – tai riba, kurią numato tam tikras modelis. Pavyzdžiui, jei projektų vadovas programinės įrangos bendrovėje prieš plėtrą patikrina visus projektus iš sąrašo ir yra tikimybė, kad vienas iš jų nebus įtrauktas į bendrąjį turimų projektų aprašą.

2 lentelė. Termino ir dinaminės ribos struktūrų aprašymas

Savybė	Terminas	Dinaminė riba
Ketiniai	Praėjus tam tikram laikui žaidimas sustabdomas	Leidžia kontroliuoti išteklių augimą
Motyvacija	Tokia struktūra padeda apriboti žaidimui skirtą laiką	Tokia struktūra padeda suteikti ištekliams dinaminį bruožą
Taikymo galimybės	Skirti laiko apribojimą kiekvienam žaidimo etapui (raundui). Pavyzdžiui, kiekvieno <i>Trogon</i> etapo trukmė ribojama	Paprastai išteklių augimas nebūna linijinis ir priklauso nuo įvairių savybių
Sandara	Riboto kiekio saugykla sujungta su sunaikinimo ir pabaigos sąlygos elementais. Pabaigos sąlygos elementas sujungtas su saugykla taikant etiketę „<1”	Susideda iš atsitiktinių vartų su filtru ir šaltinio mazgu, kurie yra susiję su ribotų išteklių saugykla
Dalyviai	Saugykla, sunaikinimo elementas, pabaigos sąlygos elementas	Saugykla, vartai, sunaikinimo elementas ir šaltinio elementas
Bendradarbiavimas	Saugykla veikia kaip taškų skaičiuoklis ir yra susijęs su sunaikinimo elementu, kad taškų skaičius būtų mažinamas. Kai rodomas taškų skaičius lygus nuliui, sužadinas pabaigos elementas	Saugykla sujungta su vartais. Po to eina jungtys su sunaikinimo arba šaltinio elementais, taip sukuriama norimas elgsenos modelis
Pasekmės	Keičia ekonomiką, kadangi ištekliams taikomi apribojimai	
Pavyzdžiai	<i>Trogon</i> , <i>Emo-bin</i>	<i>Trogon</i> , <i>TaskVille</i>
Susijusios struktūros	Riboto išteklių kiekio, nuosavybės ir tikimybės struktūros	-



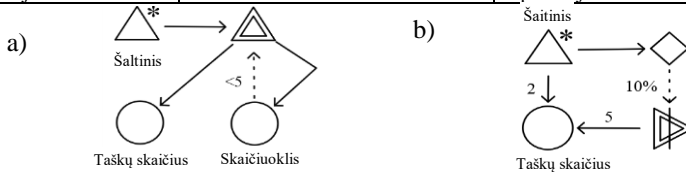
3 pav. (a) Atsitiktinio rezultato ir (b) filtravimo struktūra

Atsitiktinis rezultatas (žr. 3.a pav. ir 3 lentelę) – čia naudojamas palyginimas su lošimo kauliukų etiketėmis. Ši struktūra numato abstraktaus ryšio modeliavimą. Pavyzdžiui, „įvykdyto veiksmo vertė yra x taškų“. Tai leidžia pakeisti dalį sužaidybtos sistemos aukšto lygio abstrakcija.

Filtravimo struktūra (žr. pav. 3.b ir 3 lentelę) leidžia sumažinti surinktų taškų skaičių arba skaitiklio rodomą rezultatą esant tam tikroms sąlygoms. Filtravimo modeliai būna naudingi modeliuojant nuobaudų skyrimo taisykles sužaidybtose sistemose.

3 lentelė. Atsitiktinio rezultato ir filtravimo struktūros

Savybė	Atsitiktinis rezultatas	Filtravimo struktūra
Ketininiai	Agregacinė logika	Inversinė logika
Motyvacija	Kartais taisyklės būna pernelyg sudėtingos, kad jas būtų įmanoma sumodeliuoti, todėl būna lengviau agreguoti visa logiką į vieną kelią	Laikui bėgant ekonomika auga ir smunka. Ši struktūra leidžia sumodeliuoti ekonomikos smukimą
Taikymo galimybės	Bet kokiais atvejais, kai taisyklė galima pakeisti atsitiktiniu skaičiumi	Padeda transformuoti ar modeliuoti neigiamus žaidimo aspektus
Sandara	Du mazgai, sujungti su atsitiktinėmis jungtimis	Rankinis sunaikinimo elementas, vartai ir saugykla
Dalyviai	Jungtis ir bet kurie du mazgai	Sunaikinimo elementas, vartai ir saugykla
Bendradarbiavimas	Jungtis pereina atsitiktinį skaičių taškų	Kai vartai sužadina sunaikinimo elementą, saugykla praranda resursus
Pasekmės	Agreguoja logiką į vieną abstrakciją	Leidžia sunaikinti išteklius
Pavyzdžiai	Visi atvejai	<i>Emo-bin, Captchino, TaskVille, PowerHouse</i>
Susijusios struktūros	-	Sprendėjo struktūra



4 pav. (a) Apribojimo struktūra ir (b) išplėtimo struktūra

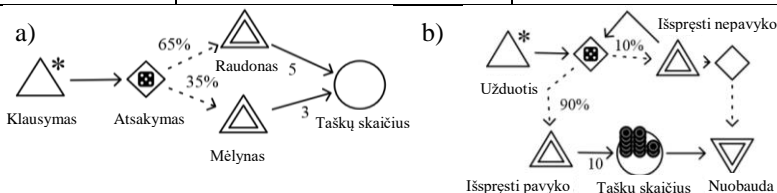
Apribojimo struktūra (žr. 4.pav. ir 4 lentelę) tam tikromis sąlygomis leidžia blokuoti kai kuriuos modelio kelius.

Išplėtimo struktūra (žr. 4.pav. ir 4 lentelę) tam tikromis sąlygomis leidžia pridėti papildomą atsitiktinį kelią. Tai padeda išplėsti įprastinį elgesį ir papildyti jį atsitiktinėmis priemokomis.

4 lentelė. Apribojimo struktūros ir išplėtimo struktūros aprašymas

Savybė	Apribojimo struktūra	Išplėtimo struktūra
Ketininiai	Tam tikromis sąlygomis leidžia kontroliuoti srautą	Padeda įtraukti tuo pačiu metu veikiančius kelius
Motyvacija	Priklausomai nuo sistemos būsenos, kartais būna naudinga apriboti arba atidaryti kelią	Kartais numatomą elgesį reikia papildyti
Taikymo galimybės	Bet kokia sistema, kur turima daug kelių (atitinkamomis sąlygomis)	Bet kokiais atvejais, kai prie numatomo kelio prisideda kitas kelias, veikiantis tuo pačiu metu
Sandara	Rankinis šaltinis sujungtas su paprasta ir būsenos jungtimi	Mazgas turi bent du kelius ir baigiasi vienu mazgu
Dalyviai	Rankinis šaltinis, saugykla	Šaltinis, vartai, vertėjas ir saugykla
Bendradarbiavimas	Kai kelias uždaromas, skaičiuoklis pasiekia reikiamo taškų skaičiaus	Kai šaltinis yra sužadintas, tuo pačiu metu aktyvuojami keli keliai

Pasekmės	Kelius galima uždaryti bei atidaryti	Prie kelio pridamas kitas kelias, veikiantis tuo pačiu metu.
Pavyzdžiai	<i>Captchino, Trogon, Meeco</i>	<i>Trogon, Captchino</i>
Susijusios struktūros	-	Nuosavybės ir tikimybės struktūra



5 pav. (a) Nuosavybės ir tikimybės struktūra bei (b) sprendėjo struktūra

Nuosavybės ir tikimybės struktūra (žr. 5.a pav. ir 5 lentelę) leidžia sukurti tuo pat metu keletą kelių arba sumodeliuoti tam tikrą vartotojo nuosavybę. Tarkime, turime sumodeliuoti kelis veiksmus to paties modelio ribose, pavyzdžiui, „Pirkiti“ ir „Pulti“. Tokiu atveju turėsime subalansuoti tarpusavyje ekonomines ir agresyvias vartotojo savybes; kuo didesnis „Pulti“ procentas, tuo agresyvesnė bus vartotojo strategija, ir atvirkesčiai.

Sprendėjo struktūra (žr. 5 pav. ir 5 lentelę) leidžia modeliuoti, kaip modelio vartotojas išsprendžia problemą. Sprendėjo struktūra leidžia sukurti vėlavimą sistemoje.

5 lentelė. Nuosavybės ir tikimybės struktūros bei sprendėjo struktūros aprašymas

Savybė	Nuosavybės ir tikimybės struktūra	Sprendėjo struktūra
Ketinumai	Modeliuojama nuosavybė ar atsitiktinė tikimybė	Leidžia modeliuoti problemos sprendimą
Motyvacija	Modeliuojamas paprastas vartotojo arba subjekto elgesys	Realiaame pasaulyje veiksmui nevyksta tuojau pat. Paprastai problemai išspręsti reikia laiko
Taikymo galimybės	Bet kur, kai norime sumodeliuoti tikimybę, kad bus įvykdytas tam tikras veiksmas arba vartotojas pasielgs tam tikru būdu	Kai norime priskirti atsitiktinį laiko tarpą tam tikrai užduočiai išspręsti
Sandara	Atsitiktiniai vartai ir keli rankiniai šaltiniai	Tai sunaikinimo elemento ir atsitiktinių vartų derinys
Dalyviai	Atsitiktiniai vartai ir rankiniai šaltiniai	Atsitiktiniai vartai, šaltiniai, vartai ir filtras
Bendradarbiavimas	Vartai atsitiktinai sužadina šaltinį	Ši struktūra jungia atsitiktinės tikimybės vartus ir šaltinį, kuris modeliuoja problemų sprendimo įgūdį. Šaltinis taip pat sujungtas su sunaikinimo elementu tam, kad būtų įmanoma modeliuoti neigiamas neteisingų sprendimų pasekmes (neprivalomas šiai struktūrai)
Pasekmės	Pasirenkamas vienas iš keleto kelių	Problemai išspręsti būna reikalingas atsitiktinis laiko tarpas

Pavyzdžiai	Visi	<i>Captchino, TaskVille, Trogon</i>
Susijusios struktūros	Sprendėjo struktūra	Sunaikinimo struktūra, nuosavybės ir tikimybės struktūra

3.3. Abstraktus formalus modelis

Analizuodami sužaidybintas sistemas, nustatėme, kad visos iširtos sistemos turėjo vieną bendrą struktūrą. Kiekviena sužaidybinta sistema yra vartotojų, taisyklių ir duomenų susidūrimas. Vartotojai sąveikauja su taisyklėmis, atlikdami tam tikrus veiksmus. Taisyklės sąveikauja su duomenimis, generuodamos turinį, kuris saugomas subjektuose. Per sąsają duomenys rodomi vartotojui. Galima teigti, kad sužaidybtoms sistemoms modeliuoti gali būti naudojamas siūlomas abstraktus modelis, kurį pavadiname UAREI (angl. User-Action-Rule-Entities-Interface). Galime naudoti UAREI modelį formaliam sužaidybtų sistemų detalizavimui, o UAREI vizualinio modeliavimo kalbą – grafiniam žaidimo mechanikos vaizdavimui. UAREI sistema gali būti naudojama visame sužaidybtų sistemų kūrimo procese.

Sužaidybinta sistema gali būti apibūdinta kaip aibė:

$$G = \{ U, A, R, E, I \}; \quad (1)$$

čia U – vartotojai, kurie sąveikauja su sistema, A – veiksmi, dėl kurių atsiranda tam tikra sistemos elgsena, R – taisyklės, kurios apibrėžia pagrindinę sistemos logiką, E – duomenų subjektai, I – sąsajos, kurios apibrėžia duomenų formatą.

Vartotojai apibrėžiami kaip aibė $U = \{ L_U, S_U \}$, čia L_U – visų išeinančiųjų nuorodų į kitus modelio elementus aibė ir S_U atrankos funkcija, kuri apibrėžia, kaip vartotojas pasirenkamas iš rinkinio įjungus imitavimo režimą.

Veiksmai yra aibė $A = \{ A_1, A_2, \dots, A_i, \dots, A_n \}$, čia A_i yra vienas veiksmas, n bendras veiksmų skaičius. Vienas veiksmas apibrėžiamas kaip $A_i = \{ L_A, S_A \}$, čia L_A yra visų išeinančiųjų nuorodų į kitus modelio elementus aibė ir S_A – atrankos funkcija, kuri apibrėžia, kaip iš rinkinio pasirenkamas su veiksmu susijęs duomenų subjektas.

Taisyklės yra aibė $R = \{ R_1, R_2, \dots, R_i, \dots, R_m \}$, čia R_i yra viena taisyklė, m bendras taisyklių skaičius. Viena taisyklė apibrėžiama kaip $R_i = \{ L_R, r_i(C, M) \}$, čia L_R yra visų išeinančiųjų nuorodų į kitus modelio elementus aibė, o $r_i(C, M)$ – taisyklės funkcija, apibrėžiama kaip:

$$r_i(C, M) = \begin{cases} \text{NULL} & \text{– jei jokios reikšmės neapskaičiuota} \\ y & \text{– jei reikšmė buvo apskaičiuota pagal taisyklę;} \end{cases}$$

čia C – dabartinio vykdymo kelio kontekstas, M – vienas iš sistemos modelių, y yra apskaičiuota rezultato reikšmė, o NULL reikšmė grąžinama, jei taisyklė netaikoma.

Taisyklės grąžinta reikšmė įrašoma į kontekstą C . Taip pat taisyklės vykdymo metu kontekstas gali būti keičiamas. Taisyklėmis tampa įmanoma kontroliuoti kontekstinį srautą sistemoje. Jei taisyklės vykdymas įvertinamas

kaip tuščias rezultatas, toks pat vykdymo kelias netęsimas toliau. Galime apibrėžti kelią „Kita“ naudodamiesi inversija „!R_i“. Jokių duomenų nebus išsaugota ir jokios kitos taisyklės nebus vykdomos, jei ankstesnės taisyklės nepavyko įvykdyti arba jei buvo gauta tuščia reikšmė, bet sistemos srautas ir toliau teiks grįžtamąjį ryšį vartotojo viršūnei. Naudojant taisykles kontekstas gali būti atnaujintas bet kuriuo būdu, kuris reikalingas užklausiai pateikti.

Subjektų aibė yra visų sistemos duomenų subjektų aibė $E = \{E_1, E_2, \dots, E_i, \dots, E_k\}$, čia E_i yra vienas duomenų saugojimo subjektas ir k yra bendras saugojimo subjektų skaičius. Vienas subjektas apibrėžiamas kaip $E_i = \{D, O, L_E\}$, čia D – subjekto schemos apibrėžimas, O – duomenų objektai, L_E – duomenų objektai.

Sąsaja yra aibė $I = \{I_1, I_2, \dots, I_l, \dots, I_l\}$, čia I_l yra viena sąsaja ir l yra bendras sąsajų skaičius. Viena sąsaja yra apibrėžiama kaip $I_l = \{L_l, Q\}$, čia L_l – visų išeinančių nuorodų į kitus modelio elementus aibė, Q – duomenų užklausa, pagal kurią pasirenkami sąsajos duomenys.



Vykdamt imitavimą bei programą, prieš pradėdant vykdyti kiekvieną veiksmą sukuriamas raktų-reikšmių rinkinys, kuris vadinamas vykdymo kontekstu. Šis kontekstas naudojamas perduodant duomenis iš vienos viršūnės į kitą. Modelio vykdymas vyksta taip: sužadintas veiksmas toliau sužadina kitas gretimas viršūnes. Sužadintos viršūnės atliks savo numatytus veiksmus ir toliau sužadins savo gretimas viršūnes. Sužadintos taisyklių viršūnės atliks savo veiksmus, duomenų esybės išsaugos reikšmes, o sąsajos atrinks duomenis, kuriuos reikia pavaizduoti vartotojui.



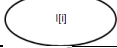

3.4. Grafinė UAREI modelio notacija

UAREI modelis yra tikslinga diagrama, susidedanti iš mazgų (viršūnių) ir nuorodų tarp mazgų (kraštų): $G = \{L, N\}$, čia N yra visų mazgų aibė, $N = \{N_1, N_2, \dots, N_i, \dots, N_m\} = U \cup A \cup R \cup E \cup I$, L yra nuorodų tarp mazgų aibė, $L = L_U \cup L_A \cup L_R \cup L_E \cup L_I$, bei L_U, L_A, L_R, L_E, L_I yra atitinkamų tipų mazgų rinkiniai, $L_X = \{L_{X_1}, L_{X_2}, \dots, L_{X_i}, \dots, L_{X_n}\}$, L_i yra nuorodų sąrašas, $L_i = (N_{out}; N_{in})$, čia $osN_{in}, N_{out} \in N$, L_{N_i} yra nuorodos, nuo kurių prasideda mazgas N_i .

6 lentelėje pateikėme grafinių simbolių sąrašą, naudojamą UAREI modelio diagramoms.

6 lentelė. UAREI modeliavimo kalbos grafinė notacija

Tipas	Simbolis	Aprašymas
Vartotojo viršūnė		Vaizduoja sistemos vartotojų grupę. Paprastai šis mazgas sužadina vieną iš galimų veiksmų
Veiksmo viršūnė		Vaizduoja veiksmą. Veiksmas sužadina išeinančias jungtis. Paprastai veiksmai būna susiję su taisyklėmis ir kitais veiksmiais

Taisyklės viršūnė		Vaizduoja taisyklės mazgą. Taisyklė apibūdina visą modelio logiką. Taisyklė sužadina kitas taisykles, subjektus ir sąsajas
Subjekto viršūnė		Vaizduoja duomenų subjektą. Kai įvyksta sužadinimas, mazgas išsaugo duomenis, kurie buvo gauti pasirinktame kontekste
Sąsajos viršūnė		Vaizduoja vartotojų sąsajas. Sužadina vartotojų viršūnes ir užbaigia grįžtamojo ryšio ciklą
Jungtis		Vaizduoja modelio tarpusavio santykius. Strėlės kryptis rodo nuo išeinančios link įeinančios viršūnės

3.5. Santrauka

Šiame skyriuje išanalizavome septynias sužaidybintas sistemas ir nustatėme bendras struktūras, būdingas dviem arba daugiau sužaidybtų programų. Buvo išskirta ir išspręsta dešimt sužaidybtų sistemų struktūrų: begalinis šaltinis, ribotas šaltinis, terminas, dinaminė riba, atsitiktinis rezultatas, nusausinimo modeliai, suvaržymas, išplėtimas, savybė ir kaita. Kiekvienas modelis turi savo motyvaciją, struktūrą, taikymo sritį ir pasekmes. Struktūros modeliuojamos *Machinations* sistemoje [28]. Šis modeliavimo įrankis leidžia greitai generuoti ir išbandyti prototipų idėjas, prieš jas įgyvendinant.

Abstraktaus formalaus modelio privalumas yra galimybė išskirti sužaidybtoms sistemoms būdingas struktūras iš daugybės įvairių formalių apibrėžimų, parašytų skirtingoms sužaidybtoms programoms. Abstraktų modelį sudaro vartotojai, veiksmai, taisyklės, duomenys ir sąsajos, būdingi visoms nagrinėtoms sistemoms. Modeliavimo kalbos sujungimas su grafų teorija leidžia suteikti informacijai paprastą, bet galingą vaizdinį pavidalą.

4. SUŽAIDYBTŲ SISTEMŲ ANALIZĖS ĮRANKIO ĮGYVENDINIMAS

4.1. Sužaidybtų sistemų plėtros metodas

Toliau aprašomas projektavimas, vykdomas taikant UAREI formalų metodą. Pradedame nuo to, kad konstruojame formalų UAREI modelį ir po to nusprendžiame, ar norime šį modelį iširti. Jei ne, viskas jau padaryta, jei taip, padarome iš turimo modelio UAREI modelį. Jei nenorime tobulinti modelio, viskas jau padaryta, jei norime tobulinti jį toliau, turime rinktis ar generuoti modelį, ar išbandyti kelias jo imitacijas ir taip modeliuoti jo veiklą. Jei nusprendėme generuoti modelį, eksportuojame darbinę programą ir ją įvertiname. Jei sistema mus tenkina, viskas jau padaryta, jei norime pagerinti sistemą, vėl konvertuojame modelį į UAREI. Jei nenorime generuoti programos, paleidžiame jos imitaciją. Įvertinę duomenis, nusprendžiame, ar norėtume atnaujinti modelį. Jei norime atnaujinti modelį, atliekame reikiamus pakeitimus ir iš naujo paleidžiame imitaciją. Jei nenorime atnaujinti modelio, patikriname, ar jis atitinka mūsų reikalavimus, ir, jei taip, vėl suteikiame jam

UAREI JSON formatą, priešingu atveju, kartojame imitacinę sesiją. Modelio ištyrimo tikslingumas ir tobulinimo tikslingumas bei kriterijai priklauso nuo eksperto sprendžiamo uždavinio ir sistemai iškeltų reikalavimų.

Mūsų sistemos pagrindą sudaro formalus UAREI modelis. Paprastai turėtumėme pradėti nuo naujo sužaidybinio sprendimo aprašymo šiame formaliajame modelyje. JSON kaip metakalbos naudojimas nėra naujas reiškinys [29], naudojamas kitose sistemose sužaidybintiems sprendimams modeliuoti ir projektuoti.

Galiausiai JSON modelį galima paversti darbine programa. Generatorius atsikrato nereikalingų modelio dalių ir sukuria darbinę programą. Paleidus tikrąją programą galima gauti duomenis, kurie gali būti naudojami toliau tobulinti sužaidybinį sprendimą.

4.2. GMOD UAREI modeliavimo ir imitavimo įrankis

Moksliniams tyrimams atlikti buvo sukurtas įrankis, kuriuo galima modeliuoti ir imituoti sužaidybinas sistemas, formaliai aprašyti naudojant UAREI modeliavimo metodą. Ši sistema bus vadinama GMOD (angl. Gamification Modeling) įrankiu.

Sistemą sudaro penkios pagrindinės dalys. Pirma, pagrindinis meniu, kuris atsako už pagrindinius programos veiksmus. Skirsnis iš kairės atsako už modelio redagavimą ir duomenų subjektų peržiūrą. Centre yra grafikų vaizdavimo priemonė, kuri parodo, kaip modelis atrodo vizualiai. Ji taip pat leidžia keisti modelį. Dešinėje yra JSON UAREI modelių rengyklė. Be kitų funkcijų, jis leidžia redaguoti modelį teksto forma. Apačioje yra pultas, kuris spausdina duomenis apie sistemos būseną.

Rengyklė turi antrą skirtuką, kuris pavadintas „Šąsajos“. Šiame skirtuke galima rasti sistemos informaciją, kuriai buvo suteiktas grafikų pavidalas. Šąsaja ir duomenų subjektų reikšmės gali būti eksportuojami CSV formatu.

Modeliavimo metu GOD suformuoja vykdomąją sistemos programą, sukuria vartotojus ir modeliuoja sistemoje jų veiksmus, grąžindama jiems grįžtamojo ryšio informaciją. Į JSON UAREI modelį buvo įtraukta papildoma meta-informacija, reikalinga modeliavimo parametrų nustatymui bei vaizdinei grafikų informacijai išsaugoti. Be to, modelyje gali būti ir pradiniai duomenys.

4.3. Santrauka

UAREI modeliavimo sistema ir jos grafinė vizualizacija buvo aprašyti remiantis abstrakčiu formaliuoju modeliu. Buvo apibrėžtas UAREI sužaidybinų sistemų kūrimo metodas. Formalusis UAREI modelis gali būti perdarytas į UAREI JSON formatą, kuris labiau tinka automatizuotai analizei. UAREI JSON galima transformuoti į vykdomąją programą.

Buvo pristatytas sužaidybinų sistemų modeliavimo įrankis (GMOD) – programa, skirta projektuoti, modeliuoti ir imituoti sužaidybinoms sistemoms; joje naudojami modeliai, išreikšti per formalus UAREI modeliavimo sistemą. GMOD naudoja UAREI JSON formatą. UAREI

modeliai gali būti transformuoti į vykdomąsias *JavaScript* programas. Įrankis leidžia modeliuoti, imituoti ir generuoti sužaidybintas sistemas. Sužaidybintų sistemų modeliavimo metodų analizės metu gautais rezultatais remtasi kuriant UAREI metodą.

5. Atskirų atvejų analizė

5.1. Trogon projektų valdymo sistema (PVS)

Paprasta projektų valdymo sistema buvo sukurta sužaidybinimo tikslais. Sistemai sužaidybinti ėmėmės šių veiksmų: 1) apibrėžti žaidimo taisykles; 2) leisti žaidėjams peržiūrėti visus darbuotojų reitingus; 3) įdiegti emblemų sistemą, kuri sudaryta iš kelių rūšių emblemų bei ženklelių lentos; 4) ženklelių sistema buvo sujungta su lygių sistema. Kiekviena emblema atstovauja tam tikram įgūdžiui, ir kuo daugiau tos pačios rūšies emblemų yra surenkama, tuo aukštesnis įgūdžių lygis yra įgyjamas; 5) ypatingi apdovanojimai ir premijos pateikiami labiausiai kvalifikuotiems darbuotojams, kaip nustatyta žaidimo taisyklėse. Pagrindiniai žaidybinimo elementai sužaidybinios *Trogon* PVS atveju – reitingas, ženklelių lenta ir projektų miškas.

Užduotys yra registruojamos ir nustatomi apdovanojimai už jų įvykdymą. Užduotys yra padalinamos priklausomai nuo to, kiek joms planuojama priskirti darbo laiko. Kiekvienas darbas gali turėti specialų įgūdžių ženklelį. Darbuotojai įveda informaciją apie savo darbo rezultatus. Kokybės inžinierius arba projekto vadovas patikrina užbaigtus darbus, užtikrindamas, kad juose nėra defektų, ir apdovanoja darbuotojus ženkleliais. Darbuotojų ženklelių ir taškų skaičius tampa matomi visiems kitiems darbuotojams. Kiekvieną savaitę išrenkamas ir apdovanojamas geriausias darbuotojas.

5.2. El. mokymosi modelis programavimo konkursui

Mokiniai iš Lietuvos mokyklų buvo pakviesti dalyvauti programavimo el. mokymosi konkurse. Šiam tikslui buvo sukurta el. mokymosi aplinka. Per aplinką mokiniai gaudavo prieigą prie mokymosi užduočių ir egzaminų. Mokymosi užduotys nebuvo privalomos vartotojams, norintiems dalyvauti konkurse. Mokiniai galėdavo gauti universiteto išduotą sertifikatą tik išlaikę egzaminą.

Laikydamas egzaminus ir atlikdamas mokymosi užduotis, vartotojas gauna taškus (0–50 už užduotį ir 0–100 už egzaminą). 10-ties pirmaujančių vartotojų taškai rodomi švieslentėje bei vartotojų reitingų lentoje. Pastaroji rodo kiekvieno vartotojo surinktus taškus ir ženklelius bei gali būti matoma bet kam. Prieš moksleiviams pereinant į kitą grupę, jie apdovanojami kelių tipų emblemomis. Siūloma sužaidybinta el. mokymosi sistema buvo išbandyta su realiais vartotojais, taip pat sumodeliuota naudojant UAREI ir GMOD įrankį.

5.3. Mažumos žaidimas

Mažumos žaidimas (angl. Minority Game, MG) yra socialinio žaidimo atmaina, kur aktyviai naudojami tiek koordinavimo, tiek konkurencijos mechanizmai [30]. MG tapo paradigma, taikoma tiriant socialinius reiškinius, kur esama daug tarpusavyje konkuruojančių veikėjų. Jei ištekliai riboti, pasirinkusiems mažumos strategiją veikėjams yra garantuota pergalė. MG buvo išsamiai tiriamas statistinės fizikos srityje, tačiau šiuolaikiniame pasaulyje tapo įprasta situacija, kad skirtingo pobūdžio ištekliai yra prieinami įvairioms socialinėms ir ekonominėms sistemoms. Mūsų tyrimui naudojamas klasikinis MG yra apibrėžtas [31].

Toliau sutelksime dėmesį į kintamojo atsipirkimo, koalicinę bei trijų komponentų MG atmainą. Sistemose, kur turima daug veikėjų, koalicijos skatina bendradarbiavimą tarp veikėjų, kuriuo siekiama pagerinti jų veiklą arba padidinti jų gaunamą naudą. Tai gali būti pritaikyta el. verslo srityje [32].

Kintamojo atsipirkimo MG (angl. Variable Payoff MG, VPMG) svarbi tiriant ką tik atsiradusias ir besivystančias elgesio formas sudėtingose realaus pasaulio socialinėse ir biologinėse sistemose, priklausančiose nuo išteklių, kurių kiekis didėja ar mažėja įvairiais būdais, kai keičiasi mažumos grupės dydis [33]. Šiuo atveju nauda gali būti įvairių rūšių, o atsipirkimas gali priklausyti nuo mažumos grupės dydžio.

Trijų komponentų balsavimo MG (angl. Ternary Voting MG, TVMG) numato, kad kiekvienam veikėjui pateikiamas trečiasis sprendimo variantas – susilaikymas. TVMG svarbi sprendimų teorijai ir gali būti taikoma politikos mokslų srityje [34]. Pasirinkęs trečiąjį variantą žaidėjas nebegali laimėti, tačiau negali ir pralaimėti. Trijų komponentų balsavimas sukuria daugiau galimybių derėtis (o tai reiškia bendradarbiavimą), aptarti visas tinkamas galimybes ir ieškoti bendro susitarimo.

Koalicinėje MG (angl. Coalition-based MG, CBMG), kai žaidėjų grupė susitaria bendradarbiauti, jie įgyja pranašumą prieš kitus žaidėjus. Mes manome, kad CBMG žaidėjų gaunamas privalumas susijęs su tuo, kad visų koalicijos narių būseną yra ta pati. Koalicijos gali būti dviejų tipų: lygios ir nelygios. Jei koalicija yra lygi, prizas paskirstomas lygiomis dalimis tarp visų koalicijos narių. Jei koalicija yra nelygi, prizas padalijamas į nelygias dalis, taip, kaip nurodyta koalicijos sutartyje. Laimi strategija, kur buvo įvesta naudingiausia koalicijos sutartis, kuri garantuoja palankiausią atsipirkimą. Žaidimas transformuojamas į aukciono žaidimą ir žaidėjai siūlo kainą vieni kitiems už palankiausią pasiūlymą.

5.4. OilTrader

OilTrader yra žaidimas, skirtas modeliuoti paskatų modelio įtaką žaidėjo sprendimui tęsti žaidimą arba iš jo išeiti. *OilTrader* yra rinkos imitavimo žaidimas, kuris leidžia iškeisti naftos akcijas į pinigus arba pirkti naftos akcijas. Šis žaidimas yra pavyzdys, iliustruojantis, kaip elgtųsi pasaulio rinkos, jei joms neturėtų poveikio jokie išoriniai veiksniai.

OilTrader yra simuliacija, kur vartotojai gali patirti supaprastintas rinkos sąlygas, prekiaudami pramanytos įmonės *OilFund* akcijomis. Žaidimo metu jie mato ankstesnius žaidimo rezultatus ir bando prognozuoti kito etapo rezultatus. Žaidimas susideda iš etapų, kurių kiekvienas trunka 15 sekundžių. Kiekvienas žaidėjas pradeda su 500 akcijų ir 500 dolerių. Kiekviename etape žaidėjas nusprendžia, parduoti ar pirkti *OilFund* akcijas; jie taip pat gali neatlikti jokių prekybos operacijų. Vieno etapo metu gali būti atlikta vienintelė prekybos operacija. Vartotojas mato keturis žaidimo skyrius. Viršuje jis mato savo pinigus ir *OilFund* akcijas. Kairiajame stulpelyje jis mato prekybos operacijų valdymo pultą ir apvalų laikmatį. Žemiau jis mato prekybos istorijos duomenis ir prekybos poveikį jo pinigams ar akcijoms.

Kiekvienas žaidėjas pradeda nuo to, kad įeina į žaidimo svetainę. Po to jis registruojasi arba prisijungia prie žaidimo. Iš pradžių žaidėjas turi pasirinkti veiksmą, kurį atliks dabartiniame etape. Jis gali išlaikyti, parduoti ar pirkti naftą. Žaidėjas pasirenka veiksmą ir įveda naftos akcijų skaičių, kurį jis nori parduoti, arba pinigų sumą, kurią jis nori išleisti naftos akcijoms. Padaręs sprendimą, jis laukia etapo pabaigos. Prekyba vertinama nustatant santykį tarp pardavėjų ir pirkėjų. Naudojant šį santykį, žaidėjų ištekliai perskirstomi remiantis mažumos žaidimo logika. Galiausiai žaidėjas gali nuspręsti palikti žaidimą arba toliau žaisti kitame etape.

6. SUŽAIDYBINTŲ SISTEMŲ ĮVERTINIMAS

6.1. Sužaidybinių sistemų įvertinimas

6.1.1. Vizualinis sužaidybinių sistemų įvertinimas

WCAG 2.0 [35] yra standartinis metodas, naudojamas tinklo sąsajos prieinamumui nustatyti. WCAG 2.0 aprašyti dviejų tipų įvertinimai: AA priskiriamas, kai kontrastas $> 4,5$, o AAA priskiriamas, kai kontrastas > 7 . Paprastai WCAG 2.0 reikalavimai naudojami tik tekstui, tačiau, kadangi šiuo atveju didžiausia informacijos dalis pateikiama vaizdais, išplėsimė šias taisykles ir pritaikysime jas paveikslėliams. WCAG 2.0 įvertinimas vyksta pagal šią schemą: jei spalvų skaičius, atitinkantis WCAG 2.0 kontrasto reikalavimus, yra didesnis nei šių reikalavimų neatitinkančių spalvų skaičius, sąsaja yra suderinama su WCAG 2.0. Jei spalvų skaičius, atitinkantis WCAG 2.0 kontrasto reikalavimus yra mažesnis nei šių reikalavimų neatitinkančių spalvų skaičius, bet ne daugiau kaip 50 %, tada sąsaja turi neįžymių problemų; jei šios problemos išsprendžiamos, sąsaja tampa suderinama su WCAG 2.0, priešingu atveju, sąsaja nėra suderinama su WCAG 2.0. Jei sąsaja yra suderinama su WCAG 2.0, tuomet įvertintos kaip AAA spalvos vyrauja, ir, priešingai, jei šios spalvos vyrauja, sąsaja bus suderinama su WCAG 2.0. Jei įvertintos kaip AA spalvos dominuoja, sąsaja yra suderinama su WCAG 2.0.

Toliau nurodyta notacija gali būti naudojama sąsajos atitikčiai aprašyti:

$$\text{WCAG 2.0 } \langle X\% \text{ AAA, } Y\% \text{ AA-}, Z\% \text{ AA} \rangle; \quad (2)$$

čia X, Y ir Z yra AAA, AA bei A vertinimus atitinkančių spalvų reikšmė procentais. Pasirinkime jau aptartą *Trogon* PVS sistemą ir pailiustruokime sužaidybintą vartotojo sąsajos įvertinimą naudojant WCAG 2.0. Buvo išanalizuoti žaidimo sluoksnių sąsajų ekrano vaizdai (JPG formatu). Eksperimentą sudaro šie etapai: 1) registruojame sąsajos vaizdą; 2) sukuriame spalvų histogramos vaizdą, naudodami *ImageMagick*; 3) naudodami *Lea Verou* įrankį, tikriname visų turimų spalvų kontrastą, lygindami su fono spalva. Šis įrankis suteikia vieną iš šių galimų vertinimų:

- jokio vertinimo nesuteikiama, jei spalvų pora nėra suderinama su WCAG 2.0;
- suteikiamas AA vertinimas, jei spalvų pora suderinama su WCAG 2.0, tik dideliems elementams;
- suteikiamas AA vertinimas, jei spalvų pora suderinama su WCAG 2.0;
- suteikiamas AA vertinimas, jei spalvų pora suderinama su WCAG 2.0 AAA.

WCAG 2.0 vertinimų rezultatai yra tokie:

- kas mėnesį atnaujinamas ženklelių puslapis, kuriame rodomas surūšiuotas visų darbuotojų sąrašas ir jų įgūdžių lygis. Ši kas mėnesį atnaujinama švieslentė su emblemomis suderinama su WCAG 2.0.

WCAG 2.0 < AAA (48 %), AA (23 %), AA-(29 %)>; (3)

- kas mėnesį atnaujinamas vertinimų puslapis, kuriame rodomas darbuotojo surinktas balas už einamąjį mėnesį. Ši kas mėnesį atnaujinama švieslentė su darbuotojų vertinimais nėra suderinama su WCAG 2.0, tačiau įmanoma pasiekti nežymaus suderinamumo;
- darbuotojų užduočių puslapis, kuriame Ganto diagramoje rodomos visos priskirtos darbuotojui užduotys. Darbuotojų užduočių puslapis nėra suderinamas su WCAG 2.0;
- kas mėnesį atnaujinamas projektų miško puslapis, kuriame rodomi visi projekto miškai, kuriuose ši mėnesį vyko kokia nors veikla. Kas mėnesį atnaujinamas projektų miško puslapis yra suderinamas su WCAG 2.0.

WCAG 2.0 < AAA (20 %), AA (36 %), AA-(44 %)>; (4)

- prietaisų skydelio puslapis, kuriame rodomos visos nebaigtos užduotys, sistemos įvykiai ir vidinės biuro žinutės. Prietaisų skydelio puslapis nėra suderinamas su WCAG 2.0;
- užduočių puslapis, kuriame rodomos visos sistemoje užregistruotos užduotys. Užduočių puslapis suderinamas su WCAG 2.0.

WCAG 2.0 <AAA(69%), AA(13%), AA-(19%)>. (5)

6.1.2. Sužaidybtų sistemų įvertinimas taikant SUS

Sistemų pritaikymo galimybių skalė (System Usability Scale (SUS)) [36] metodologiją galima naudoti siekiant įvertinti sužaidybtos aplinkos patogumą. SUS jau gali būti laikomas standartu būdų, skirtų vertinti sistemų ar produktų naudojamumą.

Paprastai SUS sudaro dešimt klausimų, kurie yra suskirstyti į penkias klausimų poras. Poroje abu klausimai yra apie tą patį, tačiau vienas užduodamas teigiamas, o kitas – neigiamas. SUS balas skaičiuojamas naudojant šią metodiką: už kiekvieną atsakymą skiriama nuo 0 iki 4 taškų. Taškų skalė yra nuo 1 iki 5. Taškai kiekvienam klausimui apskaičiuojami atėmus 1 iš pasirinktos reikšmės skalėje. Taškų skaičius klausimams, kurių numeris nelyginis, (1, 3, 5, 7, 9) yra nuo 0 iki 4. Taškų skaičius klausimams, kurių numeris lyginis, (2, 4, 6, 8, 10) yra nuo 4 iki 0. Galutinis rezultatas gaunamas padauginus taškų skaičių iš 2.5. Bendras SUS taškų skaičius yra nuo 0 iki 100.

30 dalyvių užpildė klausimyną. Pagrindinės grupės respondentų amžius buvo nuo 18 iki 35 metų. Klausimyną pildė 22 vyrai ir 8 moterys. 23 iš 30 dalyvių tiesiogiai dirbta su informacinių technologijų (IT) sistemomis.

Atsakymas į kiekvieną SUS klausimą vertinamas nuo 0 iki 10 taškų ir kiekvienas elementas susijęs su dviem klausimais. Visos sistemos sužaidybinimas gali gauti nuo 0 iki 100 taškų.

Sužaidybtos sistemos taikymo galimybės tarp lyčių skiriasi nedaug. Vidutinis skirtumas yra 3.5 taško. Tai rodo, kas lytis beveik neturi įtakos sužaidybtų sistemų praktiniam taikymui. Todėl *Trogon PVS* žaidybinimą galima suprasti ir įvertinti beveik nedarant jokio skirtumo tarp moterų ir vyrų. Nėra pakankamai duomenų statistiniam reikšmingumui teigti.

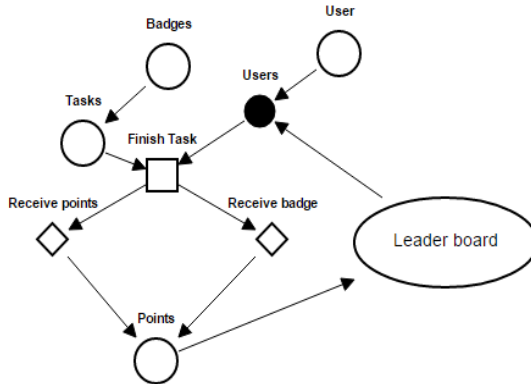
Skirtumas tarp sužaidybtų sistemų naudojimo galimybių įvertinimų, kurį nulėmė vartotojų darbo patirtis su IT sistemomis, sudaro 17 taškų. IT specialistai įvertino *Trogon PVS* žaidybinimą 75 taškais, t. y. yra 3,9% daugiau nei vidutinis įvertinimas. Studentų ir moksleivių tyrimo atsakymai rodo, kad rezultatas nėra statistiškai reikšmingas.

Visa sužaidybtina *Trogon PVS* buvo įvertinta 71 taškais iš 100. Todėl ji gavo „gerą“ įvertinimą praktinio taikymo atžvilgiu. Analizuojant tam tikrų elementų taikymo galimybes bei jų įvertinimus, švieslėntės buvo įvertintos kaip lengviausios suprasti, o žaidimų elementai - kaip sunkiausi suprasti.

6.2. Trogon PVS sužaidybinimo modelis

6.2.1. Trogon UAREI modelis

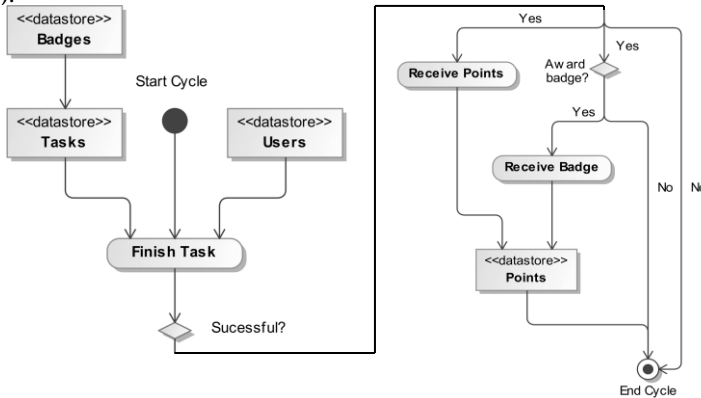
Sužaidybtų sistemų modeliavimui iliustruoti buvo pasirinkta *Trogon* Projektų valdymo sistema (PVS). Pav. 6 UAREI modelis yra pavaizduotas grafiškai naudojant siūlomą grafinę notaciją.



6 pav. Sužaidybinto Trogon PVS vizualinis modelis.

6.2.2. Trogon UML modelis

Palyginimo dėlei pateikta ML diagrama, atitinkanti tą patį loginį srautą (7 pav.).

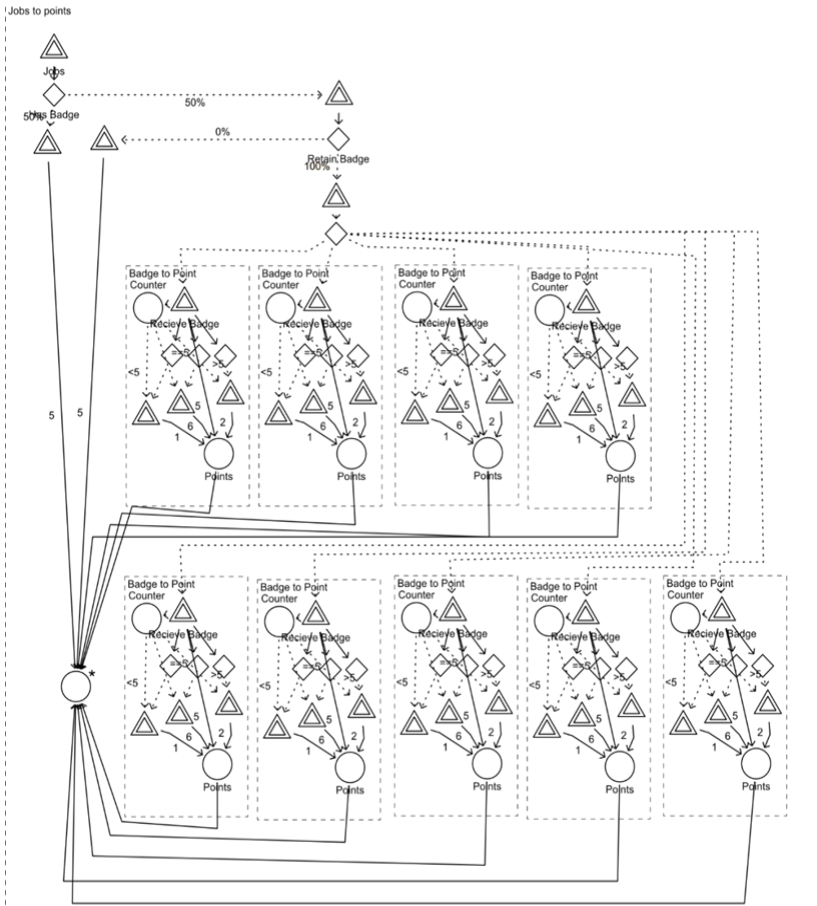


7 pav. Sužaidybinto Trogon PVS modelis, detalizuotas UML veiklos diagrama

Štai kaip veikia UML modelis. Prasidėjus kiekvienam etapui, vartotojas gali užbaigti užduotį, su kuria yra susijusi emblema. Jei darbuotojui tai pavyksta, jam suteikiami taškai, jei darbuotojas nusipelno emblemas, jis ją gauna. Rezultatai išsaugomi duomenų bazėje. Jei darbuotojui nepavyksta atlikti užduoties ar nenusipelno emblemas, ciklas užbaigiamas.

6.2.3. Trogon modeliavimas Machinations

Palyginimo dėlei tas pats modelis buvo aprašytas naudojant Machinations vaizdinę notaciją (žr. 8 pav.).



8 pav. Sužadymbinto *Trogon* PVS modelis, detalizuotas naudojant *Machinations*

Machinations sudaro trys dalys: užduočių ir emblemų generatorius, emblemų skaičiuoklis ir taškų rezervas. Sudėtingumas atsiranda dėl dviejų modelio dalių:

- norint ženklelius perteikti taškais reikia sudėtingos mazgo konfigūracijos, kad būtų galima modeliuoti tikėtiną žaidimo taisyklę, o tai rodo, kad sunku formuluoti sudėtingas taisykles;
- toks pat ženklelius atitinkančių taškų apskaičiavimas turi būti kartojamas kiekvienam iš devynių emblemų tipų, o tai rodo, kad modelis negali naudoti realaus gyvenimo informacijos.

6.2.4. Modelių palyginimas

Kadangi UAREI modelis aprašomas naudojant tam tikrų diagramų teorijos elementus, diagramų duomenis galima naudoti ir tada, kai reikia įvertinti UAREI vizualinį sudėtingumą: mazgų skaičių N , nuorodų skaičių E ir McCabe ciklo matinį sudėtingumą (McCabe Cyclomatic Complexity).

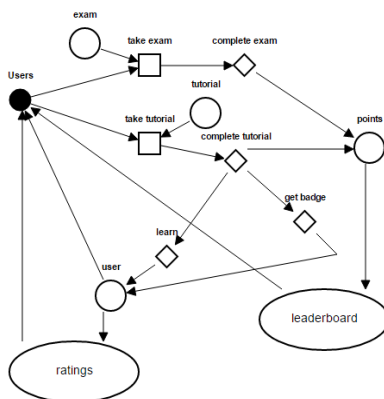
7 lentelė. Trogon PVS modelio vizualinis sudėtingumas

Sudėtingumo matmuo	UAREI modelis	Machinations modelis	UML veiklos modelis
Mazgų skaičius	9	90	11
Nuorodų skaičius	10	153	13
McCabe ciklo matinis kompleksiškas	3	65	4

UAREI ir Machinations sužaidybintų Trogon PVS modelių sudėtingumas apibendrintas 7 lentelėje. Palyginimas rodo, kad UAREI modelis yra žymiai mažiau sudėtingas, nei jo Machinations atitikmuo.

Pažintinių matmenų sistema (Cognitive Dimensions Framework, CDF) [37] - požiūris, dažnai naudojamas vaizdinėms kalboms įvertinti. Buvo atliktas išanalizuotų kalbų CDF vertinimas. Pagrindiniai UAREI privalumai yra realus pasaulio duomenų struktūros, į vartotoją orientuotas požiūris ir būdingas šiai sistemai grįžtamojo ryšio ciklas. E-mokymosi aplinkos sužaidybinimo modelis

6.3. El. mokymosi aplinkos UAREI modelis



9 pav. Hibridinis sužaidybinės sistemos modelis dalyvių įsitraukimui skatinti

Programavimo konkurso atveju modeliavimas gali būti atliekamas naudojant UAREI (angl. Users, Actions, Rules, Entities, and Interfaces – vartotojai, veiksmai, taisyklės, subjektai ir sąsajos). Šis modelis pavaizduotas 9 pav. Formalus aprašymas atitinka modelį. Bet koks vartotojas atsitiktinai pasirenka ar laikyti kokį nors egzaminą, ar spręsti mokomąsias užduotis.

Aktyvesni vartotojai bus aktyvesni ir naudodami sistemą. Mokymasis apsiriboja tik mokomosiomis užduotimis.

Dėl to vartotojų įgūdžių lygis didės laikui bėgant. Grįžtamojo ryšio ciklas atsiranda tuomet, kai vartotojas gauna informaciją iš sistemos per švieslentę ir reitingus.

6.3.1. Hibridinio sužaidybinios sistemos modelio veiklos modeliavimas

Prielaidos. Tarkime, kad yra 100 studentų, tarp kurių programavimo žinių aibė yra pasiskirsčiusi nuo 0 % iki 15 % (didesnis procentas reiškia geresnes programavimo žinias). Jeigu vartotojas užbaigia pamoką, jo žinios atsitiktinai padidės 1–5 %. Didesnė įgūdžio reikšmė padidina tikimybę, kad studentas gaus geresnį rezultatą iš mokomosios užduoties ar egzamino. Mokinių sąveika su sistema išreiškiama taškais: kuo didesnis taškų skaičius, tuo labiau studentas užsiėmęs. Manoma, kad žaidybinimas padidina naudotojų įsitraukimą, remiantis patikrintais rezultatais kitose sistemose. Kuo didesnis studento įsitraukimo rodiklis, tuo daugiau įtakos jam turi žaidybinimas. Kiekvienas egzaminas suteikia vartotojui 0–100 taškų, o kiekviena pamoka (vadovėlis) – 0–50 taškų.

Hipotezė. Daugiau įsitraukusių moksleivių grupė pralenks mažiau įsitraukusių grupę. Yra dvi grupės – mažiau ir daugiau įsitraukę sužaidybinios sistemos naudotojai. Paimkime labiau įsitraukusius naudotojus, kurie surinko daugiau nei vidutinį taškų kiekį iš mokymosi užduočių. Modeliavimo rezultatai yra statistiškai reikšmingi ir rodo, kad mažiau įsitraukusios grupės vidutinis egzamino balas yra $55,9 \pm 15,7$ %, o daugiau įsitraukusios – $65,2 \pm 13,6$ %. Perstatymo testas rodo, kad rezultatas statistiškai reikšmingas ($p = 0,727$).

6.3.2. El. mokymosi aplinkos žaidybinimo modelio eksperimentinis įvertinimas

Eksperimentai buvo atliekami internetu, pateikiant programavimo kurso sistemą moksleiviams. Sistema suteikia galimybę dalyvauti interaktyviame įvade į C ++ programavimo kalbos kursą. Sprendimus įvertina tikri dėstytojai, kurie taip pat yra tos pačios programavimo kurso sistemos registruoti vartotojai. Vertinimo diapazonas yra nuo 0 iki 100, kur 0 yra blogiausias rezultatas, o 100 – geriausias. Kiekviena išspręsta problema apdovanojama 1 tašku. Kiekvienam vartotojui taškai sumuojami. Geriausių studentų švieslentė rodoma internetinių kursų sistemos puslapyje. Sistema turi ženklelių sistemą, kuri padeda atskirti studentus, kurie yra surinkę skirtingą taškų skaičių. Studentai gali užsidirbti iki 30 taškų už papildomą sprendimą.

Studentai buvo suskirstyti į dvi grupes: kontrolinę grupę ir eksperimentinę grupę, kuri turi galimybę spręsti papildomas problemas. 95 studentai buvo atrinkti dalyvauti eksperimente. „Sužaidybiną“ grupę (studentus, kurie buvo išsprędę papildomas problemas) sudarė 48 studentai. Kontrolinę grupę sudarė 47 studentai. Buvo palyginti „sužaidybinios“ ir

kontrolinės grupės rezultatų vidurkiai. Vidutiniškas taškų skaičius už papildomą užduotį ir vidutinis „sužaidybintos“ grupės balas buvo įvertinti siekiant nustatyti studentų skaičių, kurie naudojami sužaidybinta sistema.

6.3.3. Eksperimento rezultatai

„Sužaidybintos“ grupės balo vidurkis buvo $83,13 \pm 23,26$. Vidutinis kontrolinės grupės rezultatas buvo $66,83 \pm 29,89$. „Sužaidybintos“ grupės vidutinis balas buvo 16,3 aukštesnis. Perstatymo testo tikimybė yra 56 %, todėl atmetama hipotezė apie tai, kad abu duomenų rinkiniai turi tą patį vidurkį, taigi rezultatas yra statistiškai reikšmingas. Tikimybinis skirstinys rodo, kad tikimybė „sužaidybinimo“ grupei yra mažesnė, vadinasi, sistema ne visiems veikia geriau. Atkreiptinas dėmesys į bimodalinį kontrolinės grupės pasiskirstymą.

Rezultatai mažiau ir daugiau įsitraukusioms grupėms. Mažiau įsitraukusi grupė turi vidutinį egzamino balą, lygų $79,7 \pm 24,7$, o daugiau įsitraukusios grupės vidurkis yra $90 \pm 18,97$. Rezultatai nėra statistiškai reikšmingi pagal perskirstymo testą ($p = 0,455$). Nors rezultatai nėra statistiškai reikšmingi, jie vis dėlto rodo panašius rezultatus, kaip ir tie, kurie buvo gauti modeliavimo metu. Tikimybinis skirstinys rodo, kad didesnis įsitraukimas neleidžia prognozuoti geresnių egzamino rezultatų.

6.4. Mažumos žaidimų (angl. Minority Game, MG) modeliavimas UAREI

6.4.1. UAREI pritaikymas MG

Bendrą vartotojų elgesį gali papildyti veikėjų elgesys. Veikėjas pirmas pasirenka veiksmą, naudodamas a_{pick} funkciją. Tam, kad pritaikytume strategiją dabartinei modelio būsenai, nustatome a_{key} funkciją, kuri generuoja atminties raktą su nuoroda į dabartinę situaciją. Pasibaigus ciklui, veikėjas gauna atskambinimą $a_{feedback}$ funkciją, kad jo pasirinkimas būtų įvertintas. Subjektas E_{agent_state} išsaugo visus duomenis, susijusius su veikėjais.

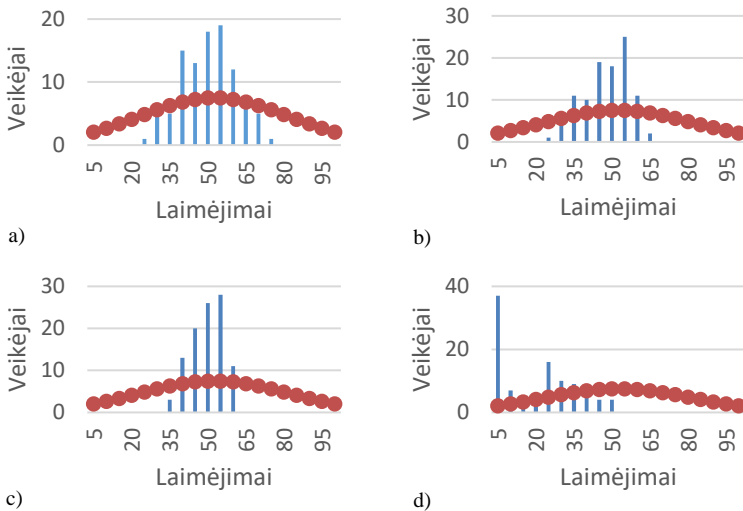
Klasikinio MG atveju galime detalizuoti šį veikėjo aprašymą taip:

$$\alpha_i = \{a_{pick}(model), a_{key}(model), a_{feedback}(model), E_{agent_state}\} \quad (5)$$

6.4.2. Veiklos modeliavimas ir rezultatai

Modeliavimui naudojama $N = 101$ veikėjui su $S = 2$ skaičiumi strategijų ir su $M = 4$ atminties apimtimi. Veikėjų veiksmas žymimi kaip +1 ir -1. Modeliavimo rezultatai rodo, kiek veikėjų priėmė sprendimą „eiti į barą“ arba „likti namuose“. Jei visų veikėjų grupės veiksmų visumos reikšmė vienam žaidimo etapui yra lygi arba mažesnė nei 50, tada ši grupė yra mažuma ir laimėjo etapą. Modeliavimas buvo atliekamas su įvairiomis MG modelių atmainomis, tarp jų ir su klasikiniu MG, bei buvo stebimas jų elgesys,

išreikštas laimėjimo funkcija, kuri čia apibrėžiama kaip laimėjimų santykis su bendroju žaidimų skaičiumi (procentais).



10 pav. Laimėjimų histograma modeliuojant (a) klasikinę MG, (b) kintamojo atsipirkimo MG, (c) koalicinę MG ir (d) trijų dalių balsavimo MG

10a pav. matome laimėjimo funkcijos histogramą po 100 žaidimo etapų skirtingose klasikinio MG imitacijose. Matome, kad laimėjusių veikėjų skaičius atitinka Gauso tikimybių pasiskirstymą (žr. standartinio, vidurkio ir Kolmogorovo-Smirnovo (KS) normalumo testo reikšmes 9 lentelėje). Laimėjimų pasiskirstymas kintamojo atsipirkimo MG pateiktas 10b. pav. Koalicinio MG atveju modeliavimo metu sistemoje atsiranda koalicija iš 20 narių. Laimėjimų pasiskirstymo histograma (10c pav.) rodo nedidelį nukrypimą nuo atsitiktinės kaitos sėkmės rodiklio (žr. 9 lentelę). Todėl galima daryti išvadą, kad koalicijos kaip metažaidimo strategijos įtraukimas į klasikinį MG modelį leidžia pagerinti kai kurių žaidėjų rezultatus. Imituojant trijų komponentų balsavinį MG (TVMG), laimėjimų histogramoje (10d pav.) matome dvi viršūnes, kurios atitinka veikėjų grupę, kurių rezultatai prasti, bei veikėjus, kurių rezultatai geri, tą ir patvirtina KS normalumo testo rezultatai (žr. 9 lentelę).

Norėdami įvertinti kiekvienos MG atmainos įdomumą, naudojome laimėjimo funkcijos neigiamos entropijos (negentropijos) reikšmę. Informacijos teorijos ir statistikos srityse negentropija naudojama matuojant atstumą nuo normalumo. Monetos metimo, paprasčiausio ir mažiausiai įdomaus žaidimo, neturinčio jokios strategijos, rezultatai turėtų Gauso pasiskirstymą. Kita vertus, bet kokie žaidimai, kur tikimybė laimėti yra vienoda arba pastovi, yra tokie pat neįdomūs. Taigi žaidimas, kuris daugiau skiriasi nuo Gauso pasiskirstymo negentropijos prasme su tuo pačiu vidurkiu

ir dispersija, gali būti laikomas įdomesniu. Tokios entropijos samprata besiremiančios priemonės jau buvo naudojamos apibrėžiant įdomumą ir duomenų staigmenos sąvokas, tarp jų ir algoritminius nulinės sumos žaidimus [38]. Analizuojamų MG atmainų laimėjimų statistinės analizės rezultatai pateikiami 9 lentelėje.

9 lentelė. Skirtingų *Minority Game* atmainų statistinis įvertinimas

MG atmaina	Vidurkis	Standartinis nuokrypis	Paklaida	KS testas	Entropija	Negentropija
Klasikinis	49.95	10.78	-0.04	1	0.932	0.02
Kintamojo atsipirkimo	48.02	8.87	-0.41	1	0.838	0.03
Koalicionis	49.75	6.50	-0.29	1	0.710	0.03
Trijų dalių balsavimo	20.74	14.82	0.32	0	0.844	0.16

Laimėjimo reikšmės klasikinio, kintamojo atsipirkimo ir koalicionio MG atmainoms yra normaliai pasiskirsčiusios, asimetrija yra priimtina (paklaida tarp -2 ir 2). Trijų dalių balsavimo modelis nukrypsta nuo normalumo dėl žaidimo taisyklių, kurios numato, kad žaidėjai, kurių rezultatai prasti, bus pašalinami iš žaidimo. Negentropijos reikšmė, kuri naudojama žaidimo įdomumui įvertinti, rodo, kad klasikinis MG, kuris turi paprasčiausią taisyklių rinkinį, yra mažiausiai įdomus, o trijų komponentų balsavimo MG, kur vartotojai gali susilaikyti arba bankrutuoti ir palikti žaidimą, yra įdomiausias.

6.5. *OilTrader Game* eksperimentas

6.5.1. Eksperimento organizacija ir tikslas

Eksperimento hipotezė teigia, kad galima įvertinti apdovanojimų mechanizmo poveikį žaidimo trukmei, priklausomai nuo skirtingų žaidėjų psichologinių tipų. Vartotojai atsitiktinai suskirstyti į dvi grupes: pagrindinę (eksperimento) grupę ir kontrolinę. Žaidimo vartotojo sąsaja kontrolinėje grupėje turi švieslentę, kuri atspindi žaidėjų pasiekimus ir parodo žaidėjo poziciją, grynąją vertę ir tai, ar vėlyviausiame žaidimo etape šis žaidėjas laimi ar pralaimi. Žaidimo vartotojo sąsaja eksperimentinėje grupėje turi papildomus tris rodiklius (juostelę, didžiausią laimėjimą ir didžiausią pralaimėjimą).

Eksperimentas turi tris pagrindinius tikslus: patvirtinti eksperimento dalyvių motyvaciją pagal pasiūlytą motyvacijos modelį bei HEXAD žaidėjų tipologiją [39] ir klausimyną [40]; nustatyti bet koki skirtumą tarp naujos, padidinto motyvavimo žaidimo sąsajos veiksmingumo eksperimentinėje ir kontrolinėje grupėse; aptarti santykį tarp HEXAD žaidėjų tipų ir žaidėjų motyvaciją tęsti žaidimą ilgiau.

6.5.2. Eksperimento subjektai

Eksperimentas buvo atliktas 2016 birželio mėnesį. Naudodamiesi microworkers.com, suformavome užduotį dalyvauti žaidime, o po to –

užpildyti žaidėjo tipo klausimyną. Kai dalyviai susikūrė paskyras, jie buvo atsitiktinai suskirstyti į kontrolinę ir eksperimentinę grupes. Žaidime dalyvavo 114 žaidėjų. Dauguma (88,6 %) dalyvių buvo vyrai. 70,2 % dalyvių buvo 20–30 metų. 17,5 % dalyvių buvo 30+ metų. 12,3 % dalyvių buvo jaunesni nei 20 metų. Dauguma (67,5 %) dalyvių žaidžia žaidimus iki 3 valandų per dieną, o 23,7 % dalyvių žaidžia daugiau nei 3 valandas. Tik 8,8 % dalyvių nežaidžia kompiuterinių žaidimų reguliariai. 69,2 % dalyvių patiko veikla, palyginus su 31,8 %, kurie pasakė, kad jie užduotį atliko tik už pinigus.

Visi žaidėjai galėjo pradėti žaisti žaidimą ir išėiti iš jo bet kuriuo metu. Kai dalyviai baigė žaidimą, jų buvo paprašyta užpildyti HEXAD klausimyną [42]. Klausimyną pildyti nebuvo privaloma, tačiau tyrime buvo analizuojami tik žaidėjų, savanoriškai užpildžiusių klausimyną, rezultatai (99 žaidėjų, 86,8 %).

Siekdami įvertinti motyvaciją gerinančius žaidimo sąsajos aspektus, mes naudojome HEXAD žaidėjų tipų klasifikaciją [40], kurioje žaidėjai skirstomi į 6 tipus: *noriai bendraujančiuosius*, *laisvadvasius*, *pirmūnus*, *filantropus*, *žaidėjus* ir *griovėjus*. Pirmiausia priskyrėme žaidėjo tipą kiekvienam žaidėjui, o tada įvertinome kiekvieno žaidėjo žaidimo trukmę. Norėdami nustatyti žaidėjo tipą, naudojome HEXAD klausimyną [39], [40].

6.5.3. Rezultatai

Šiame eksperimente rėmėmės hipoteze, kad skirtingi *OilTrader* žaidimo paskatų modeliai nevienodai veikia skirtingus žaidėjų tipus. Motyvacija tęsti žaidimą vertinama priklausomai nuo žaidimo etapų skaičiaus, kuriuose dalyvavo žaidėjas. Analizuodami atsakymus į klausimyno klausimus, pastebėjome, kad kai kurie žaidėjai užpildė jį atsitiktinai. Pagal [41] pateiktas rekomendacijas, atlikome dviejų etapų statistinę analizę. Pirmajame etape tikrinamas žaidėjų patikimumas. Šio etapo tikslas yra sukurti pseudopatikimą grupę žaidėjų, kuri bus analizuojama antrajame etape. Nepatikimų žaidėjų reitingai nustatomi remiantis HEXAD klausimyno rezultatais. Tik patikimų žaidėjų rezultatai naudojami tolesnei analizei.

Statistinės analizės etape pašalintų žaidėjų skaičius buvo 34 iš 99 (34,3 %). Tas pats žaidėjas galėjo būti priskirtas keliems skirtingiems žaidėjų tipams. Vidutinis sutapimas tarp žaidėjų grupių buvo lygus 20,5 %. Geriausiu atveju buvo tik 9,1 % panašumas (tarp *pirmūnų* ir *filantropų*), o blogiausiu atveju buvo ~ 44,4 % panašumas tarp dviejų skirtingų žaidėjų tipų grupių (*griovėjų* ir *noriai bendraujančiųjų*).

Norint įvertinti žaidimo trukmę mediana buvo pasirinkta kaip statistinė priemonė, atsparesnė išskirtims nei aritmetinis vidurkis. Vidurkio rezultatai rodo, kad eksperimentinėje grupėje žaidėjai sužaidė po $12,2 \pm 2,9$ etapų o kontrolinėje grupėje žaidėjai sužaidė po $10,3 \pm 2,4$ etapų. Suporuotų imčių t-testas atmetė hipotezę, kad šie du duomenų rinkiniai turi vienodą vidurkį ($p = 10^{-48}$).

Sukeitimo testo rezultatai rodo, kad pagrindinės grupės žaidėjams buvo didesnė tikimybė ($p = 0.671 \pm 0,008$) žaisti ilgiau negu kontrolinės grupės žaidėjams ($p = 0,329 \pm 0,008$).

Pirma, naudodami savirankos (angl. Bootstrap) metodą ($N = 1000$), apskaičiavome kiekvieno žaidimo etapo standartinius nuokrypius ir nustatėme ribas, kur pasikliautini intervalai nesutampa. Rezultatai rodo, kad žaidėjai iš pagrindinės grupės buvo labiau linkę pasitraukti iš žaidimo nuo 17 iki 25 etapo. Antra, naudodami *Students t-testą* nustatėme ribas, kur testas atmetė hipotezę, kad abu duomenų rinkiniai turi tą patį vidurkį. Rezultatai rodo, kad žaidėjai iš pagrindinės grupės buvo labiau linkę išeiti iš žaidimo nuo 7 iki 44 etapo.

Siekdami nustatyti šių klausimų statistinę svarbą pritaikėme savirankos metodą ir *Students t-testą*. Rezultatai rodo, kad svarbiausi klausimai telkiasi į konkurencingumą (Q7, $p = 0.003$), smalsumą (Q9, $p = 0.003$), naujumą (Q14, $p = 0.004$), savanaudiškumą (Q16, $p = 10^{-33}$, Q25, $p = 10^{-44}$, Q30, $p = 0.02$), autonomiją (Q18, $p = 10^{-94}$, Q22, $p = 10^{-88}$), savarankišką efektyvumą (Q20, $p = 10^{-25}$), meistriskumą (Q27, $p = 10^{-79}$) ir empatiją (Q28, $p = 10^{-52}$). Visi šie veiksniai turėjo teigiamą poveikį žaidimo trukmei. Šie rezultatai atitinka apsisprendimo teoriją [41], kuri pabrėžia autonomijos ir kompetencijos vaidmenį palaikant motyvaciją žaisti.

Modifikuota sąsaja su papildomomis paskatomis lėmė ilgesnį žaidimo laiką *laisvadvasių*, *žaidėjų* ir *griovėjų* atveju, tačiau *norai bendraujančiųjų*, *filantropų* ir *pirmūnų* atveju ji nebuvo veiksminga.

Sukeitimo testo rezultatai parodo tikimybę, kad pagrindinė grupė turės geresnį rezultatą (ilgesnį žaidimo trukmę) nei kontrolinė grupė, priklausomai nuo tipo, kuriam priklauso žaidėjai. Labiausiai reikšmingas poveikis pasireiškė *žaidėjams* ($p = 0,6906 \pm 0,008$), *laisvadvasiams* ($p = 0,6267 \pm 0,008$) ir *Griovėjams* ($p = 0,5688 \pm 0,008$).

6.5.4. UAREI MG papildymas motyvacija

Siekiant modeliuoti Minority Game, veikėjai buvo įtraukti į UAREI modelį kaip vartotojų elgsenos dalį.

$$\alpha_i = \{a_{pick}(model), a_{key}(model), a_{playing}(model), a_{feedback}(model), E_{agent\ state}\} \quad (6)$$

Modelį reikia papildyti vartotojo psichologiniu sprendimų priėmimo procesu:

$$a_{playing}(model) = \begin{cases} true, & \text{if } m_i(model) > 0 \\ false, & \text{if } m_i(model) \leq 0 \end{cases} \quad (7)$$

Čia turime skaitinę funkciją $m_i(model)$, kuri skaitmenine forma atspindi vartotojo motyvaciją. $m(model)$ funkciją galima pasirinkti laisvai. Mūsų modeliavimo tikslais naudosime tokią formą:

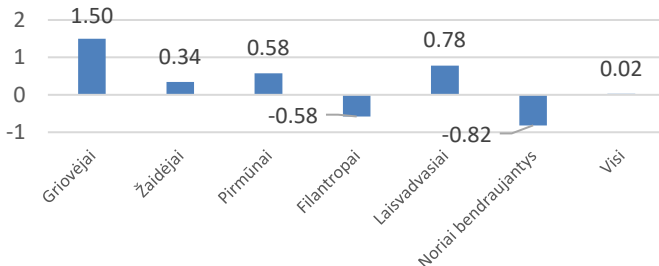
$$m_i(model) = m_{i-1}(model) + \sum_n^N O_{i,n} S_n W_n e^{-\frac{i-s_n}{\tau_n}} - S \quad (8)$$

Čia kiekvienas naujas motyvacijos rezultatas priklauso nuo ankstesnės motyvacijos būsenos. $O_{i,n} \in \{-1,0,1\}$ – koeficiento rezultatas, priklausantis nuo modelio vykdymo, S_n – skaliarinė reikšmė, atitinkanti faktoriaus svorį (priklausantį nuo žaidimo). $W_n \in [-1,1]$ – skaliarinė reikšmė, atitinkanti faktoriaus svorį (priklausantį nuo žaidėjo). i yra apvalusis indeksas. $e^{-\frac{i-s_n}{\tau_n}}$ apibrėžia, kaip kiekvieno koeficiento poveikis mažėja laikui bėgant, s_n ir τ_n reikšmės apibrėžia, kaip greitai mažėja eksponentinio koeficiento poveikis. S apibrėžia kaip greitai vartotojas praranda susidomėjimą žaidimu ir gali būti pasirinktas laisvai, tačiau siūlome naudoti šią formulę:

$$S = \frac{\sum_n^N e^{-\frac{s_n}{\tau_n}}}{K} \quad (9)$$

6.5.5. Veiklos modeliavimas ir rezultatai

Eksperimentui bus naudojamas OilTrader UAREI modelis. Šio eksperimento hipotezė – įmanoma dirbtinai imituoti atsitiktinių vartotojų elgseną ir išanalizuoti rezultatus pagal žaidėjų tipą, remiantis klasifikavimo rezultatais. Laikysime, kad vartotojus gali veikti tik jiems matomi elementai. Naudosime formulę (8) su šia konfiguracija. Kontrolinėje grupėje yra 3 ($N = 3$), o eksperimentinėje grupėje 6 ($N = 6$) koeficientai, turintys įtakos vartotojų elgesio pokyčiams. W_n – kiekvieno žaidėjo koeficientas yra atsitiktinis skaičius tarp -1 ir 1. $O_{i,n}$ gali būti lygus -1, jei šio koeficiento poveikis yra neigiamas, arba lygus 1, jei poveikis teigiamas. $O_{i,n}$ yra lygus nuliui, jei pokyčių nėra. Pagal šį modelį s_n ir τ_n yra lygūs 4. Modeliavimo tikslams imsime, kad koeficiento svoris S_n lygus 1 visiems koeficientams. Remiantis siūloma formule $S = 2e$. $m_0(model)$ yra atsitiktinis dydis nuo 30 iki 60. Simuliacija vyksta tol, kol visi žaidėjai nusprendžia liautis. Naudodamiesi Marczewski žaidėjų tipais, priskirsime imitacijoje dalyvaujančius žaidėjus labiausiai tinkantiems tipams.



11 pav. Skirtumai tarp žaidėjų grupių pagal žaidėjo tipą

Ekspirimente dalyvavo dvi grupės – eksperimentinė ir kontrolinė. Kiekvienos grupės rezultatai buvo apsvarstyti pagal atskirus žaidėjų tipus bei visiems tipams esant kartu (mišrusis tipas). Kai vidurkiai nagrinėjami neskirstant žaidėjų į tipus, negalima išvelgti jokio skirtumo tarp kontrolinės ir eksperimentinės grupės. Jei žaidėjai skirstomi į tipus, matome, kad kiekvieno žaidėjų tipo elgesys skiriasi dėl eksperimentinėje grupėje atsiradusių pokyčių.

11 pav. rodomi kiekvieno žaidėjų tipo elgesio pokyčiai. Eksperimento metu buvo pastebėta padidėjusi motyvacija *griovėju*, *žaidėju*, *laisvadvasių* ir *pirmūnų* atveju bei sumažėjusi motyvacija *filantropų* ir *noriai bendraujančiųjų* atveju. T-testas rodo, kad rezultatai nėra statistiškai reikšmingi, vadinasi, modelis nebuvo teisingai sukalibruotas.

6.6. Rezultatų aptarimas ir išvados

Pirmasis sužaidybintų sistemų įvertinimo metodas numato, kad bus naudojama WCAG 2.0 spalvotų rodiklių santykio analizė, kuri gali atspindėti sužaidybtos aplinkos vartotojo sąsajos vizualinį patrauklumą. Antrasis būdas – pritaikyti *SUS* klausimyną vartotojų atsiliepimams rinkti ir sužaidybtos sistemos patogumui įvertinti.

Buvo nustatyta, kad UAREI vaizdinio sudėtingumo laipsnis yra mažesnis. Mažesnis vizualinis sudėtingumas rodo, kad modelis yra suprantamesnis ir lengviau taikomas [42]. UAREI modeliavimo metodo privalumai yra šie: aukštas abstrakcijos lygis, metodo ypatybė, leidžianti palaikyti grįžtamojo ryšio ciklus, galimybė paversti modelį į vykdomąjį kodą, aiški riba tarp duomenų ir kodo, į vartotoją orientuotas požiūris. Hibridinio sužaidybtos sistemos modelio tyrimo rezultatai rodo, kad žaidybinimas padidina įsitraukimą lyginant su nesužaidybtą sistemą naudojusia grupe, o tai gerina moksleivių egzaminų rezultatus.

Pasiūlytas stiprinamasis modelis buvo sukurtas vieno žaidėjo žaidimams ėjimais su begaline teleologija pagal daugiadimensinę žaidimų topologiją [43] ir gali būti taikoma tiek pradedantiesiems, tiek vidutiniškai pažengusiems žaidėjams.

Klasikinio MG modelio rezultatai yra tarsi atsitiktinis monetų metimas, vadinasi, žaidimas tikriausiai nebus įdomus žaidėjams ilgesnį laiką. Kintamojo atsipirkimo MG suteikia veikėjui galimybę užsidirbti daugiau taškų vieno žaidimo etapo metu ir daro žaidimą įdomesnį. Trijų dalių balsavimo MG suteikia trečiąją pasirinkimo galimybę, taip pat žaidėjo bankroto galimybę kaip vieną iš galimų galutinių žaidimo rezultatų. Koalicionis MG modelis praturtina žaidimo taisykles, kadangi žaidėjams suteikiama galimybė derėtis tarpusavyje ir taip į žaidimo scenarijų įtraukiamas rinkai būdingas elgesys.

OilTrader žaidimo eksperimento rezultatai rodo, kad yra statistiškai reikšmingas skirtumas tarp skirtingų žaidėjų tipų ir būdų, kaip priimami sužaidybtos sistemos siūlomi motyvacijos skatinimo mechanizmai. Analizuojamas vartotojo sąsajos sprendimas (švieslentė, rodanti žaidėjų pažangą) buvo veiksmingas būdas pratęsti žaidimo trukmę kelių tipų

žaidėjams – *laisvadvasiams, griovejams* ir *žaidėjams* (pagal HEXAD tipologiją [40]), tačiau *norai bendraujančiųjų, filantropų* ir *pirmūnų* atveju motyvacijos stiprinimo efektas pasiektas nebuvo. Buvo pasiūlytas būdas imituoti sužaidybintų sistemų vartotojų motyvaciją.

Siūlomi vertinimo metodai leidžia spręsti tokias sužaidybintų sistemų problemas: (a) sistemų naudojamumo skalė ir svetainių prieinamumo gairės leidžia įvertinti sužaidybinės sistemos naudojamumą; (b) žaidėjų pergalių pasiskirstymo analizė leidžia įvertinti sužaidybintų sistemų įdomumą ir (c) sužaidybintų sistemų analizė pagal psichologinius žaidėjų tipus leidžia įvertinti kaip sužaidybinės sistemos pokyčiai veikia skirtingų žaidėjų motyvaciją ir sudaryti žaidėjų elgsenos modelius, skirtus sistemų analizei.

7. IŠVADOS

1. Svarbu įvertinti sužaidybinas sistemas tiek kiekybiškai, tiek kokybiškai. Vizualinės sužaidybintų sprendimų sąsajos gali būti vertinamos kiekybiškai, naudojant pasiūlytą metodą, grindžiamą *Interneto prieinamumo gairėmis* (WCAG 2.0), taip pat kokybiškai, naudojant modifikuotą SUS klausimyną (patvirtinti taikant *Trogon PVS* sistemai). Bet kokie pastebėti vartotojų elgsenos pokyčiai, sukelti sistemos žaidybinimo, turi būti patvirtinami statistiniais metodais.
2. Buvo išskirta dešimt sužaidybinoms sistemoms būdingų struktūrų: begaliniai ištekliai, riboti ištekliai, terminas, dinaminė riba, atsitiktinis rezultatas, filtravimo struktūros, apribojimas, išplėtimas, nuosavybė ir tikimybė bei sprendėjas. Buvo identifikuotos abstraktas sužaidybintų sistemų modelio UAREI (angl. User-Action-Rule-Entity-Interface – formalusis vartotojų, veiksmų, taisyklių, subjektų, sąsajų modelis) sudėtinės dalys: vartotojai, veiksmai, taisyklės, subjektai ir sąsajos. Šis modelis gali būti naudojamas vaizdiniam sužaidybintų sistemų detalizavimui, abstrakčiam šioms sistemoms būdingų struktūrų aprašymui, darbinių sužaidybintų sprendimų modeliavimui bei sužaidybintų programų generavimui. Vykdomųjų programų modeliavimui buvo sukurtas *Žaidybinimo modeliavimo* (GMOD) įrankis, kuris leidžia paversti UAREI modelį vykdomąja programa.
3. Palyginus su kitais žinomais sprendimais, UAREI pasižymi panašiomis arba geresnėmis galimybėmis modeliuoti sužaidybintų sistemų veiklą ir pateikti panašius modeliavimo rezultatus. UAREI modeli galima detalizuoti, modeliuoti ir prognozuoti vartotojo elgseną, išreikštą atsitiktinių imčių forma bei susijusių su realia programa (sužaidybinės sistemos dalimi), taip pat įvertinti žaidybinimo poveikį. Buvo sukurtas ir pademonstruotas metodas, leidžiantis imituoti ir įvertinti sužaidybinės sistemos poveikį vartotojo motyvacijai, atsirandantį dėl motyvacijos skatinimo

metodo, priklausomai nuo psichologinio žaidėjo tipo. Siūlomi modeliai ir metodai buvo patikrinti modeliuojant rinkos veikėjų elgesį *Minority Game* modeliu (*OilTrader* žaidimas).

LITERATŪROS ŠARŠAS

- [1] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, 2011, pp. 9–15.
- [2] J. Anderson and L. Rainie, "The future of Gamification. Pew Research Center," *Washington, DC Retrieved from <http://www.pewinternet.org/2012/05/18/the-future-of-gamification>*, 2012.
- [3] Gartner Research, "Gartner Says By 2015, More Than 50 Percent of Organizations That Manage Innovation Processes Will Gamify Those Processes," *Gart. Inc.*, p. 2015, 2012.
- [4] P. Herzig, K. Jugel, C. Momm, M. Ameling, and A. Schill, "GaML-A modeling language for gamification," in *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, 2013, pp. 494–499.
- [5] A. Mora, D. Riera, C. Gonzalez, and J. Arnedo-Moreno, "A literature review of gamification design frameworks," in *Games and Virtual Worlds for Serious Applications (VS-Games), 2015 7th International Conference on*, 2015, pp. 1–8.
- [6] B. J. Fogg, "A behavior model for persuasive design," in *Proceedings of the 4th international Conference on Persuasive Technology*, 2009, p. 40.
- [7] M. Wu, "Gamification from a company of pro gamers," *Lithosph. Community*, 2011.
- [8] B. Gnauk, L. Dannecker, and M. Hahmann, "Leveraging gamification in demand dispatch systems," in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, 2012, pp. 103–110.
- [9] F. F.-H. Nah, Q. Zeng, V. R. Telaprolu, A. P. Ayyappa, and B. Eschenbrenner, "Gamification of education: a review of literature," in *International Conference on HCI in Business*, 2014, pp. 401–409.
- [10] M. Sailer, J. Hense, H. Mandl, and M. Klevers, "Psychological Perspectives on Motivation through Gamification.," *IxD&A*, vol. 19, pp. 28–37, 2013.
- [11] S. W. Bench and H. C. Lench, "On the function of boredom," *Behav. Sci. (Basel)*, vol. 3, no. 3, pp. 459–472, 2013.
- [12] M. Csikszentmihalyi and D. K. Bose, "Flow: e Psychology of Optimal Experience." .
- [13] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun, "Boredom, engagement and anxiety as indicators for adaptation to difficulty in games," in *Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era*, 2008, pp. 13–17.
- [14] R. M. Martey *et al.*, "Measuring game engagement multiple methods and construct complexity," *Simul. Gaming*, p. 1046878114553575, 2014.
- [15] E. Gamma, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.
- [16] B. Kreimeier, "The case for game design patterns." 2002.
- [17] S. Bjork and J. Holopainen, "Patterns in game design (game development series)," 2004.
- [18] S. Kelle, R. Klemke, and M. Specht, "Design patterns for learning games," *Int. J. Technol. Enhanc. Learn.*, vol. 3, no. 6, pp. 555–569, 2011.
- [19] J. T. Kim and W.-H. Lee, "Dynamical model for gamification: Optimization of four primary factors of learning games for educational effectiveness," in *Computer Applications for Graphics, Grid Computing, and Industrial Environment*, Springer, 2012, pp. 24–32.

- [20] H. Park, "Relationship between motivation and student's activity on educational game," *Int. J. Grid Distrib. Comput.*, vol. 5, no. 1, pp. 101–114, 2012.
- [21] K. M. Kapp, "Gamification Designs for Instruction," *Instr. Theor. Model. Vol. IV Learn. Paradig. Educ.*, p. 351, 2016.
- [22] J. Tenzer, "Improving UML design tools by formal games," in *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, 2004, pp. 75–77.
- [23] G. A. Marsan, "New paradigms towards the modelling of complex systems in behavioral economics," *Math. Comput. Model.*, vol. 50, no. 3, pp. 584–597, 2009.
- [24] N. Thillainathan, "A Model Driven Development Framework for Serious Games," 2013.
- [25] S. K. Bista, S. Nepal, N. Colineau, and C. Paris, "Using gamification in an online community," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, 2012, pp. 611–618.
- [26] K. T. Chan, I. King, and M.-C. Yuen, "Mathematical modeling of social games," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, 2009, vol. 4, pp. 1205–1210.
- [27] G. W. de Oliveira, S. Julia, and L. M. S. Passos, "Game modeling using workflow nets," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011, pp. 838–843.
- [28] J. Dormans, "Machinations Diagram Tutorial," *Portf. Joris Dormans*, 2013.
- [29] A. Giurca and E. Pascalau, "JSON Rules.," in *KESE*, 2008.
- [30] J. Linde, J. Sonnemans, and J. Tuinstra, "Strategies and evolution in the minority game: A multi-round strategy experiment," *Games Econ. Behav.*, vol. 86, pp. 77–95, 2014.
- [31] D. Challet and Y.-C. Zhang, "Emergence of cooperation and organization in an evolutionary game," *arXiv Prepr. adap-org/9708006*, 1997.
- [32] L. He and T. R. Ioerger, "Combining bundle search with buyer coalition formation in electronic markets: A distributed approach through explicit negotiation," *Electron. Commer. Res. Appl.*, vol. 4, no. 4, pp. 329–344, 2006.
- [33] Y. Li, A. VanDeemen, and R. Savit, "The minority game with variable payoffs," *Phys. A Stat. Mech. its Appl.*, vol. 284, no. 1, pp. 461–477, 2000.
- [34] D. S. Felsenthal and M. Machover, "Ternary voting games," *Int. J. Game Theory*, vol. 26, no. 3, pp. 335–351, 1997.
- [35] L. G. Reid and A. Snow-Weaver, "WCAG 2.0: a web accessibility standard for the evolving web," in *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*, 2008, pp. 109–115.
- [36] J. Brooke, "SUS-A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.
- [37] T. R. G. Green and M. Petre, "Usability analysis of visual programming environments: a 'cognitive dimensions' framework," *J. Vis. Lang. Comput.*, vol. 7, no. 2, pp. 131–174, 1996.
- [38] J. Schmidhuber, "Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes," in *Anticipatory Behavior in Adaptive Learning Systems*, Springer, 2009, pp. 48–76.
- [39] G. F. Tondello, R. R. Webbe, L. Diamond, M. Busch, A. Marczewski, and L. E. Nacke, "The Gamification User Types Hexad Scale," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, 2016, pp. 229–243.
- [40] A. M. L. E. N. L. Diamond G. F. Tondello and M. Tscheligi, "The HEXAD Gamification User Types Questionnaire: Background and Development Process," in *Workshop on Personalization in Serious and Persuasive Games and Gamified Interactions*, 2015.
- [41] T. W. Calvert, E. W. Banister, M. V Savage, and T. Bach, "A systems model of the

- effects of training on physical performance,” *IEEE Trans. Syst. Man. Cybern.*, no. 2, pp. 94–102, 1976.
- [42] D. C. Donderi, “Visual complexity: a review.,” *Psychol. Bull.*, vol. 132, no. 1, p. 73, 2006.
- [43] E. Aarseth, S. M. Smedstad, and L. Sunnanå, “A multidimensional typology of games.,” in *DIGRA Conf.*, 2003.

DARIAUS AŠERIŠKIO PUBLIKUOTŲ DARBŲ SĄRAŠAS DISERTACIJOS TEMA

Straipsniai mokslinės informacijos instituto duomenų bazės („Web of Science“)

1. Ašeriškis, D., & Damaševičius, R. (2017). Computational Evaluation of Effects of Motivation Reinforcement on Player Retention, *Journal for Universal Computer Science. Graz : Technische Universitaet Graz*. 2017, vol. 23, iss. 5, p. 432-453.
2. Damaševičius, R., & Ašeriškis, D. (2017). Visual and Computational Modelling of Minority Games. *TEM JOURNAL*, 6(1), 108-116.

Konferencijų pranešimų medžiagoje paskelti straipsniai

1. Ašeriškis, D., & Damaševičius, R. (2014). Gamification patterns for gamification applications. *Procedia Computer Science*, 39, 83-90.
2. Ašeriškis, D., & Damaševičius, R. (2014). Gamification of a project management system. In *ACHI 2014: The Seventh International Conference on Advances in Computer–Human Interactions* (pp. 200-207).
3. Ašeriškis D., & Damaševičius, R. (2017). Player type simulation in gamified applications, *Proceedings of the IVUS International Conference on Information Technology*. *CEUR Workshop Proceedings*, Vol. 1856, 1-7.
4. Ašeriškis D., Tamošaitis J. *Kitokia PĮ kompanija, Mag&Doc IT2012: Kaunas, 2012*, p. 49-52.
5. Ašeriškis D., *Projektų valdymo sistemos žaidimizavimas, IVUS 2013: Kaunas, 2013*, p. 13-16.

Straipsniai kituose leidiniuose, turinčiuose citavimo indeksą

1. Maskeliūnas, R., Gudonienė, D., Ašeriškis, D., Blažauskas, T., Vasiljevas, M., & Drėgvaitė, G. (2016). Hybrid eLearning Model for Increasing Learner’s Engagement. *Transylvanian Review*, (1).
2. Ašeriškis, D., Blažauskas T., & Damaševičius, R. (2017). UAREI: a model for formal description and visual representation of software gamification. *Journal of the Facultad de Minas, Universidad Nacional de Colombia-Medellin Campus*, Vol. 84, Issue 200, 326-334.

Sąrašo sudarymo data: 2017-09-01

INFORMACIJA APIE AUTORIŲ

Išsilavinimas:

- 2007-2011 Įgijo informatikos inžinerijos bakalauro kvalifikacinį laipsnį Kauno technologijos universitete, Informatikos fakultete.
- 2011-2013 Įgijo informacijos saugos magistro kvalifikacinį laipsnį Kauno technologijos universitete, Informatikos fakultete.
- 2013-2017 Studijavo Kauno technologijos universiteto doktorantūros studijose (informatikos inžinerija 07T).

Darbo patirtis:

- 2017 birželis – iki dabar: Inžinerijos direktorius, KAYAK Software Corporation;
- 2017 sausis - 2017 birželis: Inžinerijos vadovas, KAYAK Software Corporation;
- 2013-2017 rudens semestrais: lektorius, Kauno technologijos universitetas;
- 2014 rugsėjis - 2017 sausis: Komandos vadovas, KAYAK Software Corporation;
- 2014 kovas – 2014 rugsėjis: Vyresnysis programinės įrangos inžinierius, KAYAK Software Corporation;
- 2012 birželis - 2014 sausis: Grafinių sąsajų programuotojas, Modnique;
- 2009 rugsėjis – 2012 birželis: programuotojas, Baltic Condition.

El. paštas: daraser@gmail.com

RÉSUMÉ

Motivation

Recently, gamification has gained popularity in the development of enterprise information and e-commerce systems. Gamification is the use of elements of game design (game rules, game techniques, gamified interfaces) in non-game contexts, such as marketing, employee performance and training, and innovation management.

Game domain is the closest domain to the Gamification domain. Gamified systems can be viewed as simpler versions of games. According to the Webster Dictionary, a game is a physical or mental competition conducted according to rules. In many cases, gamification can be easily applied to games and vice versa.

The modelling of gamification is important for the design of systems based on the principles of serious games, in order to quantify and validate the impact of gamification and to better understand why and how gamification works. Existing evaluations of gamification usually focus on the application of user questionnaires and other methods of qualitative evaluation. There is still a lack of modelling methods and tools to aid the design and development of gamification in serious systems.

This dissertation aims to introduce modeling and simulation methods which would allow us to build a bridge between the formal modeling of gamification and quantitative simulation of games, analysis and evaluation of game rules and processes.

Object and scope of the research

The object of the research is methods and tools for simulation, analysis, and evaluation of gamified software systems. These methods and tools are necessary if we want to employ more powerful game patterns, elements and mechanics into our systems at the same time as understanding how the system will impact user behavior.

The scope of the research involves:

- Methods for game modeling, analysis, prototyping.
- Formal mathematical modelling of gamified systems.
- Multi-user system user behavior modeling and model behavior analysis.
- Gamified system analysis and evaluation.

Problem statement and research questions

The problem of this work focuses on the lack of methods and tools for quantitative analysis and understanding of gamification patterns, elements and mechanics. The same problem is observed in the related game domain. Currently, there is no single integrated process which would lead game

designers from the idea of gamification to the final implementation of a gamified system.

This dissertation gives answers the following research questions:

- What is state-of-the-art in the domain of gamification modelling?
- How can gamified systems be evaluated?
- How can gamified systems and elements of gamification be modeled abstractly?
- How can user behavior in gamified systems be predicted?

Aim and objectives

The aim of the research is providing the gamification domain with a tool and methods for modeling, analyzing, simulating and generating gamified systems, isolating patterns, and understanding gamification pattern impact on user behavior.

For the aim of the thesis to be achieved, the following objectives have been set out:

1. To conduct static and dynamic analyses of gamified systems to identify methods for evaluation of the system gamification.
2. To consider gamified systems for patterns and identify commonalities in gamified systems, and create a gamification modelling method.
3. To examine known solutions for evaluation of usability and efficiency of gamification solutions, and create a method for analyzing and computationally modeling the impact of gamification on the behavior of users with respect to gamified systems.

Defended propositions

1. The proposed gamification evaluation methods can be used for quantitative and qualitative evaluations of gamified systems.
2. The proposed visual gamification modelling method allows for creating gamified system models, extracting gamification patterns, simulating user behavior, analyzing simulated user behavior, comparing gamified systems and generating gamified applications.
3. The UAREI model simulation in the GMOD tool reproduces similar behavior of other tools and enables to predict how an implemented prototype gamified system will act in a real-world environment. The efficiency of the gamification solutions can be modelled by analyzing the behavior of users using a game player type evaluation based on the HEXAD questionnaire.

Major contributions and novelty of the research

Major contributions of this work:

- Formal abstract gamification modeling method as a common method for analyzing gamified systems has been created. This method allows us to model, examine and evaluate gamified systems has been developed, which has similar or better capabilities versus current industry methods.
- A new method for evaluating gamified system attraction was created by comparing player winning distribution to normal distribution. Method tested by analysis of minority game variation.
- A new method for evaluating gamification reinforcement models by psychological player types was proposed. Method tested by analysis of OilTrader game experiment.
- Proposed methods for gamified system user interface evaluations using Web Content Accessibility Guidelines and System Usability Scale, tested with Trogon project management system.
- The proposed methods and gamified system modelling method allows game designers and scientists to develop gamification models, simulate gamified systems, improve models to achieve desired outcomes and generate systems from models has been created. This increases the development speed of gamified systems.

The novelty of the method:

- UAREI is a new formal modeling method dedicated for gamified system modeling with visualization and simulation capabilities.
- The created method allows different ways to simulate gamified applications. For simulation UAREI utilizes custom selection functions, agent based modelling and Minority Game engine.
- Simulating gamified systems allows to evaluate gamified system performance in new ways like: increase of motivation, results or game interestingness.
- Created gamified system evaluation methods based on usability, visual attractiveness (contrast ratios), and player motivation by player types.

Practical significance

- The proposed formal abstract model enables scientists to formally model gamified systems. Using the same formal notation extraction of common gamification patterns and pattern composition into new systems (UAREI is mathematically expressed through sets which allows easy mathematical manipulation using algebra of sets) becomes easier.
- Simulation of the gamified system model gives valuable insights on how the gamified system affects user behavior to game designer and scientists. Simulation provides faster feedback and predicts

what kind of behavior might be expected from users interacting with the gamified systems.

- Software generation from UAREI creates the shortest path from gamified system models to a prototype system ensuring fast iteration over the gamified system solution which allows for better results.
- The analysis of the gamified systems using the UAREI modeling method helps to better understand how gamified system patterns effect end-user behavior, how patterns interact with each other and how to achieve best gamified systems performance, before deploying ready-to-use gamified software systems.

Scientific approval

The results of the research have been presented in two international conferences, and two articles have been published in journals indexed in Web of Science Journal List. One article is still in review in the journal included in Web of Science Journal List. One paper is published in a peer-reviewed journal. Three topic-related papers have been presented in Lithuanian conferences. Two papers have been presented in international conferences in Spain and France. The full list of publications can be found in chapter titled “LIST OF PUBLICATIONS OF DARIUS AŠERISĖKIS ON DISSERTATION THEME”. A list of conferences:

- The 22th International Master and PHD Students Conference “Information Society and University Studies”, 2017 April 28, Kaunas;
- The Seventh International Conference on Advances in Computer-Human Interactions ACHI 2014 March 23 - 27, Barcelona, Spain;
- The Sixth International Conference on Intelligent Human Computer Interaction IHCI 2014 December 8-10, Envy, France;
- The 18th International Master and PHD Students Conference “Information Society and University Studies”, 2013 April 25, Kaunas;
- The 17th Master and PHD Students Conference “Information Society and University Studies”, 2012, Kaunas.

Structure of the dissertation

In Chapter 2, related works from various areas of gamification like motivation and psychological foundation, modeling, software architecture and agent-based simulation have been analyzed. When analysis of gamification is conducted, patterns from gamified system models are extracted. Also, the UAREI abstract formal model is identified, which is the basis for further research. GMOD is presented as an UAREI modeling tool which is used for system model analysis and simulation. In Chapter 5, case studies used for research are described.

Chapter 6 covers methods for gamified system quantitative and qualitative evaluation. Furthermore, the UAREI modeling method is evaluated against UML and Machinations. Furthermore, hybrid eLearning UAREI model simulation is evaluated against real system user behavior. UAREI is extended to support a market-based simulation adapted from Minority Game engine and evaluation of gamified system by player types, which is presented thusly. In Chapter 7, conclusions of the work are presented. Finally, references and a list of publications by the thesis author is given.

CONCLUSIONS

1. It is important to evaluate gamified systems both qualitatively and quantitatively. Visual interfaces of gamification solutions can be evaluated quantitatively using the proposed method based on the Web Accessibility Guidelines (WCAG 2.0), and qualitatively using the modified SUS questionnaire (both validated in the Trogon PMS system). Statistical methods should be applied to validate any observed change in user behavior due to the effects of gamification (such as the improvement of students' exam results for gamification solutions applied in the programming contest).
2. Ten gamification patterns have been identified: infinite source, limited source, time limit, dynamic limit, random result, drain patterns, constrain, extension, property and change, and solver. The components of the abstract gamification model (User-Action-Rule-Entity-Interface formal model (UAREI)) have been identified: users, actions, rules, entities and interfaces. The model can be used for the visual specification of the gamified systems, abstract description of gamification patterns, executable modeling of gamification solutions, and generation of gamification applications. For executable modeling, the Gamification modeling (GMOD) tool, which allows for the transformation method from UAREI model to executable application, has been developed.
3. UAREI has similar or better capabilities than other industry solutions for simulating gamified systems and providing similar simulation feedback versus other known solutions. UAREI can be used to specify, model and predict randomized user behavior with respect to a real application of the gamified system and evaluate the effects of gamification. A method for simulating and evaluating gamified system impact on user motivation by psychological player types using motivation reinforcement model has been developed and demonstrated. The proposed models and methods have been tested by simulating behavior of market agents applying the Minority Game model (OilTrader game).

UDK 519.83 + 004.414.23](043.3)

SL344. 2017-08-29, 3 leidyb. apsk. l. Tiražas 50 egz.

Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249
Kaunas

Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424
Kaunas