

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ MAGISTRANTŪROS STUDIJŲ PROGRAMA

**Kiberfizinių-socialinių sistemų reikalavimų analizė ir modeliavimas**

**Analyzing and modeling the early requirements of cyber-physical-social systems**

Magistro baigiamasis darbas

Atliko: Evaldas Naujikas (parašas)

Darbo vadovas: dr. Audronė Lupeikienė (parašas)

Recenzentas: dr. Jolanta Miliauskaitė (parašas)

Vilnius – 2020

## SANTRAUKA

Remiantis moksline literatūra, darbo analitinėje dalyje yra nagrinėjamos kiberfizinės-socialinės sistemos, jų savybės ir įvairios reikalavimų specifیکavimo metodikos, siekiant vieną jų pritaikyti šioms sistemoms. Darbo teorinėje dalyje pasiūlytas reikalavimų metodinis karkasas, reikalavimų pavyzdinis modelis, modifikuoti probleminiai freimai šių sistemų reikalavimams analizuoti ir modeliuoti, o praktinėje – atliktas sukurtos metodikos vertinimas.

**Raktiniai žodžiai:** kiberfizinė sistema, kiberfizinė-socialinė sistema, reikalavimai, reikalavimų analizė, reikalavimų modeliavimas, probleminiai freimai, SysML, FIPA.

## SUMMARY

Using results of library research, in the analytic part of this master thesis, cyber-physical-social systems, their characteristics and various requirement specification methods are being analyzed. The aim of this is to propose modifications for one of the methods for these systems. In the theoretical part of the thesis a new methodological framework is suggested to analyze and model early requirements of these systems. The experimental part of the thesis is devoted to the validation of the proposed framework.

**Keywords:** cyber-physical system, cyber-physical-social system, requirements, analysis of early requirement, requirement modelling, problem frames, SysML, FIPA.

## TURINYS

ĮVADAS .....	5
1. KIBERFIZINIŲ SISTEMŲ KLASĖS .....	9
1.1. KIBERFIZINĖS SISTEMOS SAMPRATA .....	9
1.2. KIBERFIZINĖS SISTEMOS SAVYBĖS .....	12
1.3. KIBERFIZINIŲ-SOCIALINIŲ SISTEMŲ SAMPRATA IR SAVYBĖS .....	14
2. REIKALAVIMŲ ANALIZĖ IR MODELIAVIMAS .....	19
2.1. SPRENDINIŲ GRINDŽIAMŲ REIKALAVIMAI .....	21
2.2. AGENTAIS GRINDŽIAMAS SPECIFIKAVIMAS .....	22
2.3. PROBLEMINIO DOMENO ANALIZĖ .....	23
2.4. KITI SPECIFIKAVIMO BŪDAI .....	29
3. DALYKINĖS SRITIES ANALIZĖ IR MODELIAVIMAS KURIANT KFSS .....	30
3.1. PROBLEMINIAI FREIMAI .....	34
3.2. ANALIZĖS IR MODELIAVIMO PAVYZDINIS PROCESAS .....	45
4. METODIKOS REALIZACIJA .....	47
REZULTATAI IR IŠVADOS .....	54
LITERATŪROS SĄRAŠAS .....	56

# IVADAS

## Nagrinėjamas objektas

Reikalavimų specifikuojimas gali būti apibrėžtas kaip iki šiol neegzistavusios sistemos ir jos elgsenos išradimas bei aprašymas [Bra02]. Kitaip tariant, sprendžiamas sistemos išorinio projektavimo uždavinys, išrandant naujos sistemos reiškimąsi (angl. appearance) ir elgseną. Nauja sistema turi tenkinti naudotojų poreikius ir spręsti jiems reikiamas problemas, kas ir sąlygoja būsimą sistemą. Tam reikia atrasti ir išsiaiškinti dalykinėje srityje jau egzistuojančias „sistemos dalis“ bei reikiamas spręsti problemas. Dalykinės srities analizės rezultatas – jos modelis ir aprašymas, kurie yra būtini ir pakankami būsimai sistemai specifikuoti. Turinio požiūriu analizės rezultatai apima relevantišką dalykinės srities savybių modelį ir reikiamą naujos sistemos poveikį, kaip reikiamos problemos sprendimo rezultata.

Šiandieniniam naudotojui reikia rezultatų, kuriuos gali teikti sistemos, apimančios dalykinės srities fizinių, socialinių ir skaitmeninių aspektų kombinaciją. Kitaip tariant, žmonių (plačiau, socialinių esybių) poreikius ir jų problemų sprendimą gali realizuoti tik fizines, socialines ir kompiuterines sudedamąsias dalis integruojančios sistemos.

Kiberfizinėmis-socialinėmis sistemomis (angl. cyber-physical-social system, toliau – KFSS) yra vadinamos naujos kartos sistemos, kurios integruoja skaičiavimus, fizines galimybes ir žmogaus dalyvavimą bei įvairiais būdais sąveikauja su naudotoju [ZYL+16]. Kiberfizinėse sistemose fiziniai ir programiniai komponentai yra persipynę, funkcionuoja skirtingose erdvėse ir laiko skalėse, veikia ir sąveikauja tarpusavyje keliais skirtingais būdais, kurie gali keistis atsižvelgiant į kontekstą [NSF13]. Kiberfizinėse-socialinėse sistemose atsiradęs socialinis lygmuo sujungia kompiuterio ir žmogaus intelektus į bendrą visumą. Tai reiškia, kad KFSS, pasinaudojant įvairia informacija iš kibernetinio, fizinio ir socialinio lygmens, turi atitikti žmogaus poreikius ir atitinkamai valdyti fizinį pasaulį. Taip pat jos gali keisti organizacijų funkcionavimą ir valdymą [LYW+11]. Išmanus miestas, namų bendrija, transporto srauto valdymas gali būti kiberfizinė-socialinių sistemų pavyzdžiais.

Kiberfizinės-socialinės sistemas charakterizuoja šie ypatumai [EPR+14]:

- Erdvėje išskirstytos sistemos dalys;
- Išskirstytas valdymas, priežiūra ir vadovavimas;
- Dalinai autonomiškos posistemės;
- Dinamiškas visos sistemos perkonfigūravimas skirtingose laiko skalėse;
- Susiformuojančios elgsenos galimybė;
- Nenutrūkstamas sistemos vystymasis jai funkcionuojant;
- Žmogaus elgsenos analizavimas ir poreikių išskyrimas;

- Fizinio pasaulio pritaikymas pagal žmogaus poreikius.

Šiame darbe nagrinėjamos kibernetinių-socialinių sistemų reikalavimų inžinerijos problemos, konkrečiau – dalykinės srities analizės, modeliavimo ir sistemos reikalavimų išvedimo metodikos.

## **Darbo aktualumas**

Viena pagrindinių kibernetinių-socialinių sistemų kūrimo problemų – fizinio, socialinio ir programinio pasaulių integravimas, apimantis tai įgalinančias atlikti kūrimo paradigmas, metodikas, metodus ir kalbas [ZTL+16]. Fizinio ir socialinio pasaulių įvykiai turi atsispindėti kibernetiniame pasaulyje, o jame priimami sprendimai turi būti perkelti atgal į fizinį pasaulį. Apjungiant šiuos pasaulius, daugelis sistemų pradeda funkcionuoti kitaip nei buvo numatyta jų projektavimo ir realizavimo metu (pavyzdžiui, intelektualūs namai). Šio tipo sistemose duomenys, informacija ir žinios traktuojamos kaip holistinė visuma, kuri yra interpretuojama, be to, žmonėms pateikiamos su jų kontekstu susijusios reikšmingos abstrakcijos [SAH13].

Kibernetinės-socialinės sistemos keičia sistemos sampratą, į ją įtraukiant žmones, infrastruktūrą ir kompiuterinę platformą [EPR+14]. Pagrindinės tokių sistemų kūrimo problemos yra susijusios su sudedamųjų dalių heterogeniškumu, lygiagrečiai vykstančiais fiziniiais, socialiniais ir kompiuteriniais procesais, kurie yra ir tolydiniai, ir diskretūs. Teoriniai tokių sistemų pagrindai dar tik pradėti kurti, nes:

- Prielaidos joms kurti atsirado prieš kelerius metus vystantis daiktų internetui, socialiniams skaičiavimams, proaktyvioms paslaugoms ir panašiai [EPR+14]. O sistemų įvairovė yra labai didelė;
- Šių sistemų kūrimą nagrinėja daugelio skirtingų mokslo šakų (pvz., mechanikos, elektros ir elektronikos, valdymo, informatikos) atstovai, kurie pateikia savo siūlymus ir kūrimo metodikas, atkreipdami dėmesį į jų požiūriu svarbiausius aspektus.

Kibernetinių-socialinių sistemų kūrimas yra plati, tarpdisciplininė sritis, kuri dar neturi bendrai nustatytų šių sistemų klasifikavimo standartų ir tas klases charakterizuojančių ypatumų aprašų. Sistemų kūrimo problemų sprendimas sąlygoja fundamentalių pokyčių (angl. paradigm shift) poreikį sistemų inžinerijoje. Svarbios problemos, kurios turi būti sprendžiamos ir reikalavimų inžinerijoje, apima statinių ir dinaminių dalių sinergetinio efekto nagrinėjimą, dinamišką evoliucionuojančių sistemos veikimą, daugeliu abstrakcijų grindžiamą modeliavimą [Hor14].

Apibendrinant galima teigti, kad ši sritis nėra pilnai iširta, kas pabrėžia jos svarbą ir aktualumą.

## Darbo naujumas

Kiberfizinės-socialinės sistemos yra charakterizuojamos kaip sintetinės, hibridinės, galinčios pasiekti tikslus, kurie nepasiekiami sudedamųjų dalių „igimtomis“ galimybėmis. Per paskutinįjį dešimtmetį kuriant kompiuterinius, fizinius ir socialinius aspektus integruojančias sistemas buvo suprasta, kad šio uždavinio sprendimui netinka tradicinė redukcionistinė paradigma, kuri buvo ilgą laiką taikyta moksle ir inžinerijoje [NIST13].

Išrandant naujos sistemos reikimąsi ir elgseną, reikia atrasti ir išsiaiškinti dalykinėje srityje jau egzistuojančias „sistemos dalis“ bei reikiamas spręsti problemas. Tam programų sistemų reikalavimų inžinerijoje naudojamos struktūrinės, objektinės, paslaugų ir probleminės domeno analizės paradigmos.

Struktūrinė analizė išplėtė reikalavimų inžinerijoje naudojamas kalbas, įvedus koncepcinius modelius poreikių ir reikalavimų specifikacijose [Bra02]. Tačiau ši metodika daro prielaidą, kad egzistuojanti sistema tenkina sistemos reikalavimus ir naujai kuriama sistema yra išvedama iš senosios. Taip pat paveldimas sistemos pagrindą sudarantis struktūrinis modelis.

Objektinė analizė faktiškai yra taikoma sistemai modeliuoti. Jacobson pabrėžė, kad objektinė inžinerija gali būti pradama naudoti nuo to laiko, kai sistemos specifikacija yra sukurta [Jac96]. Tam tikros objekcinio modeliavimo technikos gali prisidėti prie dalykinės srities ir jos subdomenų modeliavimo, tačiau objekcinio aspekto įnašas privalumų nesuteikia [Bra02].

Dalykinės srities dekomponavimas į paslaugas ir paslaugomis grindžiamas modeliavimas tradicinę objektinę analizę papildo naujomis veiklomis ir artefaktais [Ars04]. Šios paradigmos naudojimas leidžia atsižvelgti į faktą, kad jau pradinėje stadijoje yra skirtingus domenus atstovaujančios sistemos dalys. Be to, identifikuojant paslaugas turi būti nagrinėjami ir socialinės sistemos tikslai.

Taikant probleminę domeno analizę išreikštiniu būdu atskiriama nagrinėjama dalykinė sritis ir kuriama sistema, taip pat problemų dalykinėje srityje aprašas ir reikalavimai, kurie turi būti tenkinami [Jac01]. Vienoje iš konferencijų M. Jackson savo pranešimo metu atkreipė dėmesį, kad ši paradigma gali būti taikoma kuriant kiberfizinės sistemas, tačiau išsami analizė ir metodinės rekomendacijos pateiktos nebuvo. Taip pat nėra aišku, ar tokio tipo analizė gali būti pakankama kiberfizinė-socialinių sistemų inžinerijoje.

Visos paminėtos analizės paradigmos gali būti taikomos kiberfizinė-socialinių sistemų kontekste, tačiau jų taikymo metu gaunami rezultatai nėra užtektini. Todėl galima teigti, kad nagrinėjama problema yra nauja.

## **Literatūros pasirinkimo kriterijai**

Kiberfizinių ir kiberfizinių-socialinių sistemų savybių apibrėžimui buvo atlikta mokslinės literatūros apžvalga pagal pateiktus kriterijus:

1. Šaltinio pavadinime arba santraukoje turi būti bent du raktiniai žodžiai iš šio sąrašo: kiberfizinės sistemos, kiberfizinės-socialinės sistemos, specifikavimas, modeliavimas, ankstyvieji reikalavimai (angl. early requirements), reikalavimai, reikalavimų inžinerija, bendrija, išmani bendrija, išmanūs namai, namų bendrija;
2. Šaltinis yra cituojamas šaltinis (angl. reference).

## **Tikslas ir uždaviniai**

Magistrinio darbo tikslas – probleminę domeno analizę (angl. problem domain oriented analysis, toliau – PDOA) pritaikyti kiberfizinių-socialinių sistemų reikalavimų inžinerijos dalykinės srities analizei ir modeliavimui.

Darbo uždaviniai:

1. Apibrėžti kiberfizinių ir kiberfizinių-socialinių sistemų savybes.
2. Išanalizuoti kompiuterinių sistemų kūrime naudojamas reikalavimų analizės ir specifikavimo metodikas.
3. Pritaikyti reikalavimų analizės ir modeliavimo metodiką kiberfizinėms-socialinėms sistemoms.
4. Atlikti sukurtos reikalavimų analizės ir modeliavimo metodikos vertinimą.



## 1. Kiberfizinių sistemų klasės

Pirmame magistrinio darbo skyriuje nagrinėjamos kiberfizinės ir kiberfizinės-socialinės sistemos, pateikiamos jų sampratos, apibrėžiamos šių sistemų savybės bei pateikiami jas iliustruojantys pavyzdžiai.

### 1.1. Kiberfizinės sistemos samprata

Apie terminu „kiberfizinė sistema“ (toliau – KFS) įvardijamą sistemų klasę pradėta kalbėti prieš dešimtmetį, kai buvo suprasta, kad įvairūs daiktai su komunikavimo ir skaičiavimo galimybėmis gali iš esmės pakeisti kuriamų sistemų galimybes. Tačiau net iki šių dienų KFS dar neturi vienos ir vienodai interpretuojamos apibrėžties.

Kiberfizinės sistemos nėra visai nauja sistemų klasė. Jos išsivystė iš kelių sričių sinergijos efekto, kuris tapo galimas dėl išaugusios skaičiavimų galios, įterptinių sistemų sumažėjimo iki miniatiūrinio dydžio ir galimybės bet kur naudotis kompiuterių tinklu, dėka. Gunes ir kiti autoriai [GPG+14] išskyrė aštuonias sritis, kurios persidengia su KFS:

- Dideli duomenys (angl. big data). Didelis kiekis duomenų, kurių nepajėgia įvertinti žmonės, tradicinės statistikos technikos ir duomenų vizualizavimas;
- Debesų technologijos. Žmonės bet kada ir bet kurioje vietoje gali pasinaudoti konfigūruojamomis skaičiavimų mašinomis (angl. computing machines), kurios yra sujungtos į bendrą tinklą;
- Sistemų sistemos. Įvairių sistemų kombinacija, turinti bendrą tikslą ir jas apjungiantį funkcionalumą, tačiau kiekviena sistema yra projektuojama ir valdoma skirtingų savininkų;
- Mechatronika. Nagrinėjimas, kaip mechaniniai įrenginiai yra kontroliuojami ir valdomi elektroninių komponentų bei programinės įrangos pagalba;
- Kibernetika. Tarpdisciplininis grįžtamojo ryšio (angl. feedback loop) tyrimas techninėse, socialinėse ir neurologinėse sistemose;
- Daiktų internetas. Komercinis terminas, kuris reiškia prie interneto prisijungiančias technines sistemas;
- Daiktų saitynas (angl. web of things). Daiktų interneto papildinys, reiškiantis fizinių sistemų gebėjimą įsijungti į pasaulinį saityną įprastinėmis „web“ technologijomis;
- Mašina-su-mašina komunikavimas (angl. machine to machine communication). Daiktų interneto dalis, kur sistemos dalinasi duomenimis be žmonių įsikišimo.

R. Poovendran [Poo10] apibūdina kiberfizinės sistemas kaip perspektyvias naujos klasės sistemas, kurios giliai įtvirtina šių sistemų galimybes fiziniame pasaulyje. Tokios galimybės gali būti tiek žmonėse, tiek infrastruktūrose ar platformose. Sistemų paskirtis – pakeisti fizinio ir kibernetinio pasaulio sąveiką. Tobulėjimas kibernetiniame (pvz., komunikavimas tarpusavyje,

skaičiavimai ar kontrolė) arba fiziniame (pvz., naujos medžiagos ar atsinaujinantis kuras) pasaulyje skatina šios klasės sistemų naudojimą, dėl jų gebėjimų vertinti ir kontroliuoti situaciją, panaudojant įvairius jutiklius ir skirtingų tipų fizinius manipulatorius.

R. Baheti ir H. Gill [BGi11] apibrėžia kiberfizinės sistemas kaip naujo tipo sistemas, kurios turi integruotas ir programines, ir fizines įrangos sistemų galimybes. Šios integracijos dėka KFS sugeba bendrauti su žmonėmis naujais būdais. Sistema praplečia fizinio pasaulio galimybes, tam naudodama įvairias bendravimo formas ir skaičiavimus. Naujos kartos bepiločiai lėktuvai, autonominiai miesto automobiliai ir žmogaus mintimis valdomi protezai yra tik keli KFS pavyzdžiai.

JAV Nacionalinio mokslo fondo programoje [NSF17], skirtoje šio tipo sistemoms, minima, kad KFS – tai sistemos, kurios yra sukonstruotos iš ir kartu priklausančios nuo neįjungiamos integracijos tarp skaičiavimų realizuojamų algoritmų ir fizinių komponentų. Tobulėjant šioms sistemoms gerės jų našumas, prisitaikymas, plečiamumas, sauga, panaudojamumas ir gebėjimas atstatyti prarastas/pažeistas savybes. Visa tai iš esmės pakeis žmonių sąveiką su sistemomis. Numatoma, kad labiausiai KFS paveiks tokius sektorius kaip transporto, energetikos, žemės ūkio, medicinos, nes jos leidžia pasiekti didesnę efektyvumą ir saugumą. Aukščiau paminėtame darbe taip pat atkreipiamas dėmesys, kad daiktų internetas sudaro prielaidas vystyti šio tipo sistemas, kadangi leidžia surinkti duomenų iš įvairiausių sensorių. Planuojama, kad ateityje KFS sugebės šią gaunamą informaciją paversti veiksmis, taip automatizuojant begalę įvairiausių darbų.

Nepaisant didelio susidomėjimo ir spartaus KFS vystymosi, teorinių ir eksperimentinių tyrimų rezultatai šioje srityje dar nepakankami, todėl konstruoti aukšto patikimumo kiberfizinės sistemas kol kas yra problematiška. Pavyzdžiui, turint milijardą įvairiausių įrenginių ir sugebant visą gaunamą informaciją apdoroti, ir valdyti bei sąveikauti su žmogumi ir fiziniu pasauliu tampa be galo sudėtinga užtikrinti pasiekiamumo, patikimumo, saugumo ir privatumo reikalavimus. Kitas bendravimo lygmuo tarp kibernetinių ir fizinių komponentų reikalauja naujų architektūrinių modelių, kurie įgyvendintų naują suvokimą apie struktūras, konstrukcijas bei jų funkcijas. Jie turi integruoti pastovų veikimą ir abstraktumą, kurį nuolat veikia nežinoma, kintanti ir atvira aplinka.

J. Kim savo disertacijoje [Kim14] pabrėžia svarbias KFS savybes, nagrinėdama, kaip sumažinti (arba visai panaikinti) kiberfizinių sistemų resursų gedimo (t.y., tiek fizinės, tiek programinės įrangos) padarinius ir užtikrinti jų saugumą. Šios savybės yra ypač svarbios atominėse jėgainėse, autonominiuose automobiliuose ir panašiose sistemose. KFS privalo užtikrinti, kad tam tikra vykdoma veiksmų seka būtų pilnai atlikta nustatytu laiku. Vienas iš dabartinių sprendimų (pvz., autonominiuose automobiliuose) sujungti kelis procesorius į bendrą komponentą, kuris atliktų sudėtingus skaičiavimus. Tačiau tai nėra efektyvus ir pigus sprendimas, kadangi galiausiai gali pritrūkti vietos automobilyje įmontuoti procesorius (įsivaizduokite

automobilio duris, kuriose po 100 procesorių), o avarijos atveju žalos kaštai labai išaugtų. Kalbant apie gedimų išvengimą, darbe [Kim14] minimi įvairūs sprendimai, keli iš jų yra:

1. Programinės įrangos replika saugoma atskirame komponente ir įsijungia tada, kai per apibrėžtą laiko tarpą nėra gaunamas atsakymas iš originalios įrangos;
2. Virtualioje atmintyje yra saugoma informacija apie tuometinę sistemos būseną, tokiu būdu užtikrinant, kad replikos galės sužinoti apie įvestį ir galimą išvestį, užduoties įvykdymo baigties laiką. Jeigu sensorius nebepateikia jokios informacijos, tai gali padaryti kitas sensorius, veikiantis šalia.

Vienas iš galimų ir dažniausiai naudojamų variantų yra trigubas dubliavimas (angl. triple modular redundancy), kai lygiagrečiai veikia 3 sistemos (gali būti ir daugiau), siekiančios bendro rezultato, o galutinis rezultatas yra sujungiamas į vieną bendrą. Tokį sprendimą naudoja NASA kosminiuose laivuose. Tačiau tai reikalauja žymiai daugiau resursų, negu komponentų atjungimas palaipsniui (angl. graceful degradation), kuris tik smarkiai sulėtina sistemą, tačiau išlaiko visas funkcijas.

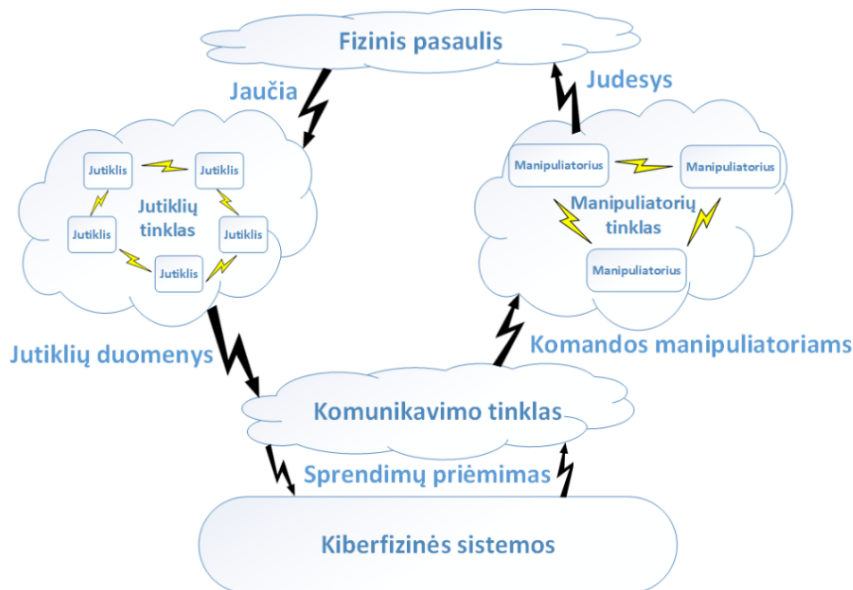
B. Schätz straipsnyje [Sch14] teigiama, kad kiberfizinių sistemų negalima suprasti tik kaip didelio mastelio įterptinių sistemų tinklo, nes jos sąlygojo sistemų kūrimo paradigmos pokyčius. Autoriaus nuomone, paprastai naudojama KFS samprata, kai jos suprantamos kaip glaudi sąveika tarp kompiuterinio ir fizinio pasaulio, neatspindi esminių šių sistemų savybių, nes toks jų supratimas apima tik įterptines (angl. embedded) sistemas. M. Broy ir kiti autoriai [BCG12] pabrėžia, jog šio tipo sistemos taip pat sudarytos iš logistinių, koordinavimo bei valdymo procesų. Paminėtos sistemos skiriasi nuo įterptinių sistemų tuo, kad KFS yra sujungtos ne vien lokaliu, bet ir globaliu tinklu, kas leidžia jas išskirti į naujo tipo sistemą. Žemiau pateikiami keli iš tokių sistemų pavyzdžių:

1. Protinga eismo valdymo sistema, kuri identifikuoja kur, kiek ir kokių objektų juda, taip optimizuodama srautą;
2. Protinga energijos paskirstymo sistema, kuri gedimo atveju identifikuoja probleminę vietą ir ją atjungia, taip sumažindama neigiamą problemos efektą.

Remiantis atlikta literatūrine apžvalga galima teigti, kad visuose nagrinėtuose moksliniuose šaltiniuose KFS yra suprantamos kaip programinės ir fizinės įrangos sistemų junginys, susietas internetu ir skirtas gauti žmogui reikiamą rezultatą. Taip pat verta paminėti, kad kiberfizinė sistema turi leisti bet kuriuo momentu žmogui perimti jos valdymą.

Magistriniame darbe analizuojamoje literatūroje pateiktos KFS apibrėžtys akcentuoja kelis skirtingus šių sistemų aspektus, tačiau visos jos pabrėžia skaičiavimų (programinė įranga, apimant įterptinę), jutiklių ir manipuliatorių (aparaturą) bei komunikavimo (tinklas) integravimo svarbą. Žemiau pateikiamas abstraktus kiberfizinės sistemos modelis (žr. 1 pav.), kuris parodo, kad fizinis

pasaulis yra suvokiamas per vieną ar daugiau jutiklių, kurie yra sujungti į tinklą. Komunikavimo tinklo pagalba KFS gauna reikiamą informaciją sprendimui priimti. Sprendimas yra paverčiamas į komandas, kurios yra perduodamos manipuliatorių tinklui jas vykdyti.



1 pav. Abstraktus kiberfizinės sistemos modelis [GPG+14]

1 pav. pateiktas modelis galėtų būti papildytas sąveika su naudotojais ir kibersocialines sistemas įgyvendinančiais elementais, nes žmonės kaip individai arba kaip socialinis vienetas veikia, ir yra veikiami techninių sistemų. Detaliau šis papildymas bus nagrinėjamas 1.3 poskyriuje „Kiberfizinės-socialinės sistemos“.

## 1.2. Kiberfizinės sistemos savybės

Kiberfizinėms sistemoms būdinga ne tik sistemų įvairovė, bet ir dėl to atsirandanti jas išskiriančių savybių gausa. KFS esminės savybės gali būti klasifikuojamos, išskiriant tris jų grupes (žr. 2 pav.).



2 pav. KFS esminės savybės [Sch14]

Pirmajai grupei yra priskiriamos savybės, nusakančios pačios sistemos funkcionalumą. Tokios funkcijos gali būti:

1. Savarankiškas prisitaikymas (angl. self-adoption), kai sistema privalo pati gebėti pakeisti savo elgesį ir struktūrą, jai norint adaptuotis (pvz., atnaujinimai) arba atsižvelgti į aplinką;
2. Savaimė pasitaisanti (angl. self-healing), kai sistema sugeba aptikti ir reaguoti į gedimus.

Į antrąją grupę patenka savybės, atsirandančios dėl persikertančių sričių. Pavyzdžiui, tokios savybės yra:

1. Tarpdisciplininės, nes atlieka įvairiausias funkcijas ir yra veikiamos daugybės išorinių procesų;
2. Tarpfunkcinės, nes geba atlikti funkcijas iš kelių sričių.

Trečioji grupė apima kritines savybes. KFS paprastai palaiko kritinius procesus, šių sistemų negalima išjungti norint jas atnaujinti, rekonfigūruoti, modernizuoti ar iš naujo įdiegti.

Kiberfizinėms sistemoms paprastai būdingos žemiau išvardintos savybės:

- Buvimas visur (angl. omnipresence). Sistema egzistuoja (beveik) visur. Dėl didėjančios įvairių įrenginių galios, viena ir ta pati sistema gali būti visuose įrenginiuose, o savo būseną dalintis tarp įrenginių naudodama debesų ar kitokias technologijas. Geriausias pavyzdys – išmanus namas, kur tą pačią sistemą galima valdyti tiek ant šaldytuvo esančio ekrano ar planšetės pagalba, tiek balsu;
- Pastovus ryšys (angl. ambient connectivity). Ši savybė leidžia naudotis ryšiu bet kuriame prietaise, bet kuriuo momentu ir nepriklausomai nuo naudojimosi priežasties ar ryšio teikėjo. Pavyzdžiui, jeigu visi mieste esantys žmonės sujungtų visus maršrutizatorius į vieną bendrą tinklą, tai turėtume didelėje miesto dalyje pastovų internetinį ryšį, kas yra itin svarbu norint išnaudoti pirmos savybės privalumus;
- Aukšto lygmens automatizavimas. Paminėta savybė lemia tai, kad kuriant sistemas svarbu užtikrinti, jog jose neatsirastų begalinių ciklų;
- Pilnai arba dalinai autonomiškos posistemės [EPR+14];
- Dinamiškas sistemos perkonfigūravimas skirtingose laiko skalėse [EPR+14] (pvz., bet kada gali atsirasti arba dingti vienas ar keli sensoriai, o duomenys iš jų gali būti gaunami kas sekundę ar kas minutę);
- Prisitaikanti elgsena [EPR+14] pagal naudotojus ir aplinką;
- Nenutrūkstamas sistemos vystymasis jai funkcionuojant [EPR+14];
- Nėra svarbus duomenų gavimo šaltinis (ar iš šaldytuvo, ar lauko termometro, ar pan.).

### 1.3. Kiberfizinį-socialinių sistemų samprata ir savybės

Vien kibernetinio ir fizinio pasaulių neužtenka, norint įvykdyti uždavinius, kuriuose sujungiamos kelios tokios pačios sistemos. Šio tikslo įvykdymui papildomai reikalingas socialinis lygmuo, kas atsispindi kiberfizinės-socialinės sistemos (toliau – KFSS) termino formuluotėje.

Mokslinėje literatūroje nėra vieno bendro sąvokos „socialinis lygmuo“ apibrėžimo. Vieni autorių darbuose teigiama, kad socialinis lygmuo yra sistemos bendravimas tik su žmonėmis (angl. human-centric), o kitų – kad tai yra bendradarbiavimas su žmonėmis arba kitomis sistemomis.

Viename iš šaltinių minima [ZYL+16], kad kiberfizinį-socialinių sistemų atsiradimas iš pagrindų pakeičia bendravimą tarp žmonių, kompiuterių ir fizinės aplinkos. Tačiau viena iš kelių problemų, kurias savo darbe nurodo šio straipsnio autorius, yra būtent socialinio lygmens atsiradimas. Šiose sistemose yra labai sudėtingi ryšiai tarp įvairiausių skirtingų dalių (tiek fiziniuose, tiek kibernetiniuose lygmenyse), taip pat nėra efektyvaus būdo jas specifikuoti ir projektuoti. SysML pagalba galima aprašyti kibernetinį ir fizinius lygmenis, tačiau visų trijų vienu bendrinio (angl. unified) modeliu aprašyti neįmanoma. Verta paminėti, kad KFSS saugumas yra dar svarbesnis ir dar mažiau ištirtas, nei KFS.

Straipsnyje [WCC+17] taip pat pritariama teiginiui, kad KFSS integruoja ne tik kompiuterius ir daiktus, bet ir žmones. Tyrėjų nuomone, šios sistemos yra ne tik kiberfizinį sistemų plėtinys, bet ir daiktų interneto plėtinys. Taigi, KFSS susideda iš daiktų interneto, socialinių, kibersocialinių ir kiberfizinį sistemų.

F. Wang savo darbe [Wan10] lygina kiberfizinės-socialinės sistemas su Karlo Poperio realybės teorija. Remiantis ja, mūsų visata susideda iš trijų tarpusavyje sąveikaujančių pasaulių:

1. Pirmasis pasaulis yra pilnas materialių objektų ir įvykių, pavyzdžiui, gamtos reiškinių;
2. Antrasis pasaulis yra pilnas protinių objektų ir įvykių, pavyzdžiui, perėjimas nuo sveikų skaičių prie realiųjų arba įsivaizduojamų skaičių atsiradimas matematikoje;
3. Trečiasis pasaulis – žmonių kuriamas pasaulis, kuriame yra visos žinios, pavyzdžiui, mitai arba mokslo pasiekimai, teoremos.

F. Wang teigia, kad pirmasis pasaulis yra identiškas, tačiau antrasis pasaulis KFSS atveju būtų kibernetinė erdvė, o trečiasis – įvairios programos (žr. 3 pav.). Autorius taip pat pabrėžia, kad šios sistemos žmonių gyvenimams turės tokią pačią įtaką, kokią matematikoje padarė įsivaizduojami skaičiai.



3 pav. Trys KFSS pasauliai [Wan10]

J. Huang, M. Seck ir A. Gheorghe [HSG16] KFSS vadina „Kiberfizinėmis-socialinėmis išmaniomis sistemomis“ (angl. Cyber-Physical-Social Smart Systems (CPS3)), atsiradusiomis daiktų interneto dėka. Autorių nuomone, fundamentalūs pokyčiai sistemos projektavime, palaikyme ir jos veikimo principu yra neišvengiami. Projektuojant šias sistemas svarbus ne tik jų patikimumas, bet ir kaip jos bus priimtos naudotojų, t.y. kaip sistemom pelnyti žmonių pasitikėjimą. Vis daugiau žmonių nešioja sportines apyrankes „Fitbit“, kurios pastoviai seka naudotojo širdies darbą, miego ritmą ir pan., o visą informaciją perduoda į įmonės serverius, kur tuos duomenis apdoroja jau kitos programos. Jeigu duomenys būtų paviešinti ar pasisavinti trečių asmenų, dauguma naudotojų šios apyrankės atsisakytų, nepaisant kūrėjų pažadų ištaisyti šią klaidą. Padarytos žalos jau nebūtų įmanoma atstatyti, o susigrąžinti žmonių pasitikėjimą būtų nelengva (ar net neįveikiama) užduotis.

Vienas iš daugelio galimų kiberfizininių-socialinių sistemų pavyzdžių – autobuso atvykimo laiko nuspėjimas, pasinaudojant keleivių mobiliais įrenginiais. Tokiu pačiu principu veikia ir „Google Maps“ žemėlapiai, kai kelionės trukmė yra apskaičiuojama pagal eismo sąlygas, kurios matomos pagal naudotojų būvimo vietą. Jeigu gatvėje labai daug naudotojų, kurie stovi arba minimaliai juda, tai šioje vietoje yra fiksuojama automobilių spūstis. Kadangi „Google“ turi daugelio gatvių realaus laiko duomenis, jų sistema gali apskaičiuoti vidutinį ruožo įveikimo laiką su kamščiais.

Kad žmonės yra KFSS dalis, o ne išoriniai veikėjai, pabrėžiama ir [LYW+11] straipsnyje. Šaltinyje teigiama, kad šios sistemos turi turėti tokias pačias žinias ir protinius gebėjimus, kaip ir žmonės. Tyrėjai prie kiberfizinės-socialinės sistemos savybių priskiria keturis domenus: fizinį, informacijos, kognityvinį, socialinį. Būtent šie domenai ypatingai svarbūs JAV karinei sistemai, vadinamai „Command and Control“<sup>1</sup>, kuri atsakinga už armijos sekimą realiu laiku. Tokio

<sup>1</sup> „Global Command and Control System“ [žiūrėta 2020-05-26]. Prieiga per internetą: <https://fas.org/nuke/guide/usa/c3i/gccs.htm>

pubūdžio sekimas turi būti labai tikslus, išsamus (pavyzdžiui, negali trūkti bent vieno kario koordinatės) ir greitas. Sistema turi gebėti realiu laiku analizuoti didžiulius duomenų kiekius, o pagal analizės rezultatus automatiškai priimti įvairiausias sprendimus, ypač kritinėse situacijose. Aprašoma sistema susideda iš įvairios techninės įrangos (angl. hardware), programinės įrangos (angl. software) ir daugelio integracijų su kitomis sistemomis. Visa tai yra sujungiama į vieną bendrą visumą, kad naudotojai galėtų patogiai ir nepertraukiamai naudotis šia sistema.

Praktinė kiberfizinių-socialinių sistemų nauda aprašoma ir darbe [RON17]. Šiame straipsnyje nagrinėjama viena iš šiuolaikiniams naudotojui aktualių problemų – transporto spūstys. Autorių teigimu, didžiausi kamščiai susidaro prie lankytinų vietų, pavyzdžiui, mokyklų, darželių, parduotuvių ar turistų lankomų vietovių. Blogiausia, kad eismo spūstys susiformuoja ne tik periodiškai, bet ir netikėtai, pavyzdžiui, dėl įvykusios avarijos arba pasimetusių vairuotojų. Norint sumažinti kamščius, reikia identifikuoti, kuri kryptis labiausiai užstrigusi ir joje prailginti žalios šviesos šviesoforo signalo veikimo laiką. Tačiau ši trukmė turi būti apskaičiuota pagal įvairius parametrus, nes per ilgai degant žalios spalvos šviesoforo signalui vienoje pusėje, kitoje pusėje susidarys nauja spūstis arba pablogins jau esamą situaciją. Taigi, tai nėra problemos sprendimas, geriausiu atveju – problemos sušvelninimas, blogiausiu – tik problemos perkėlimas iš vienos sankryžos pusės į kitą. Autorių nuomone, pasinaudojant socialiniais tinklais (pavyzdžiui, „Twitter“), galima nuspręsti, kuriose sankryžose ir kuria kryptimi yra susiformavęs kamštis. Mūsų teigimu, pasinaudoti galima ir kitais, tolimesniais šviesoforais, kurie pranešdami apie srauto intensyvumą ir jo greitį gali informuoti apie grėšiančias transporto spūstis.

Ramadhan, Oktaria ir Nugraha nurodo, kad naudojant fizinius įrenginius (šviesoforus) yra rodoma žalia arba raudona spalva, o naudojant programinę įrangą yra ieškoma socialinių įrašų, kuriuose yra išreiškiama negatyvi nuomonė ir paminėta lankytina vieta. Pagal rastus įrašus yra identifikuojama probleminė vieta ir prailginamas signalo laikas. Prailginimo trukmė priklauso nuo negatyvumo stiprumo ir nusiskundimų skaičiaus arba kol atsiranda negatyvūs pranešimai iš toje pačioje sankryžoje kita kryptimi keliaujančių naudotojų. Socialinių tinklų pagalba žmonės gali pranešti apie savo nepasitenkinimą eismu, taip vienašališkai komunikuojant su sistema.

Straipsnyje [WZY+17] nurodoma, kad KFSS turi ne tik paprastus resursus, kuriuos gali panaudoti tikslui pasiekti, bet ir paslaugas kaip resursus, t.y., kad kai kurie resursai yra visiškai nepriklausomi, o tiesiog užsakomi. Tokiu atveju nei eiga, nei kokybė negali būti kontroliuojama. Pavyzdžiui, picos gaminimas namie turint produktus yra paprastas resursų panaudojimas, o picos užsakymas į namus – paslauga kaip resursas. Užtikrinti, kad paslauga būtų atlikta, nuspėti kada tai įvyks arba įvertinti paslaugos atlikimą ir panašiai, galimybių nėra, todėl šios problemos kol kas neturi aiškaus sprendimo būdo.



Išmanių namų bendrija – dar vienas iš kiberfizinių-socialinių sistemų pavyzdžių. Jeigu įsivaizduojamame kvartale būtų vien tik išmanūs namai, jų visumą galima būtų pavadinti išmanių namų bendrija. Tokio tipo namai sugebėtų bendrauti tarpusavyje (kas atitinka socialinį lygmenį), atlikti įvairiausių darbus, pavyzdžiui, užrakinti spynas, atidaryti arba uždaryti langus (kas atitinka fizinį lygmenį), apdoroti duomenis, ateinančius tiek iš fizinio, tiek iš socialinio pasaulių, tiek iš kitos programinės įrangos (kas atitinka kibernetinį lygmenį).

Kaip teigia straipsnio [HSG16] autoriai, viena iš daugelio išmanių namų paskirčių yra patogaus gyvenimo ir sveikata besirūpinančios aplinkos suteikimas. Šios gyvenimo sąlygos ypatingai svarbios senyvo amžiaus žmonėms, kurie vis dažniau gyvena vieni. Kita tokio būsto paskirtis – namų saugumas, pavyzdžiui, automatinis langų uždarymas, durų užrakinimas, gaisro ir jo vietos identifikavimas arba likvidavimas. Be saugumo užtikrinimo gali būti ir įvairūs, komfortą suteikiantys, veiksmai, pavyzdžiui, temperatūros ir drėgmės palaikymas, pastovus apšvietimas ir panašiai. Užmaršiams žmonėms galėtų padėti įvairios pastabos, pavyzdžiui, priminimas išgerti vaistus arba atlikti kokią nors mankštą. Tai yra tik kelios iš daugelio galimų funkcijų, kurias gali atlikti išmanūs namai.

Straipsnyje [XMa11] aprašomos išmanios bendrijos, kurių apibrėžimas tinka ir išmanių namų bendrijai. Norint suprasti kas yra išmani bendrija, pirmiausia reikia atkreipti dėmesį į tai, kaip suprantama įprastinė. Įprastinė bendrija – tai grupė bendraujančių žmonių, kurie dažniausiai turi bendras vertybes, charakteristiką arba interesus. Dažnai bendrija yra suprantama kaip didelė grupė asmenų, gyvenančių šalia vienas kito. Pavyzdžiui, galima teigti, kad miestas susideda iš bendrijų rinkinio.

Internetinė bendrija yra virtuali ir egzistuojanti tik internete. Tai yra grupė žmonių, bendraujančių informacinių technologijų pagalba, dažniausiai – virtualioje erdvėje. Šiai žmonių grupei nėra svarbi narių gyvenamoji vieta. Todėl tokia grupė dažnai būna sudaryta iš skirtingose pasaulio vietose esančių žmonių.

Autorių teigimu, išmani bendrija – tai grupė tarpusavyje susijusių socialinių objektų, kurie bendrauja vieni su kitais įvairiais tinklais ir visiems savo nariams gali suteikti įvairiausias paslaugas. Šie tarpusavyje susiję objektai gali būti ne tik žmonės, bet ir stalas, raktas ar šuo. Dažniausiai tokie objektai stengiasi pasiekti vieną bendrą tikslą.

Šaltinyje minima, kad išmanios bendrijos gali pastebimai pagerinti žmonių gyvenimo kokybę, ypatingai vyresnio amžiaus asmenų. Pavyzdžiui, jos gali atlikti tokius veiksmus, kaip neįgalių ar sunkiai sergančių asmenų pastovus nuotolinis stebėjimas, tikslus žmogaus judesių sekimas ar net nuotolinė pagalba. Paminėtos galimybės gali sumažinti eiles pas gydytojus, dėl ko galima būtų daugiau dėmesio skirti skubiems ar net mirtiniems atvejams. Taip pat šios bendrijos turi galimybę tarpusavyje dalintis įvairia informacija, pavyzdžiui, išmanusis namas gali kitiems bendrijos

nariams pranešti apie įvykusį apiplėšimą, kas padėtų kuo skubiau jiems apsisaugoti (užrakinti visas duris, įspėti šeimininkus ir uždaryti visus vartus).

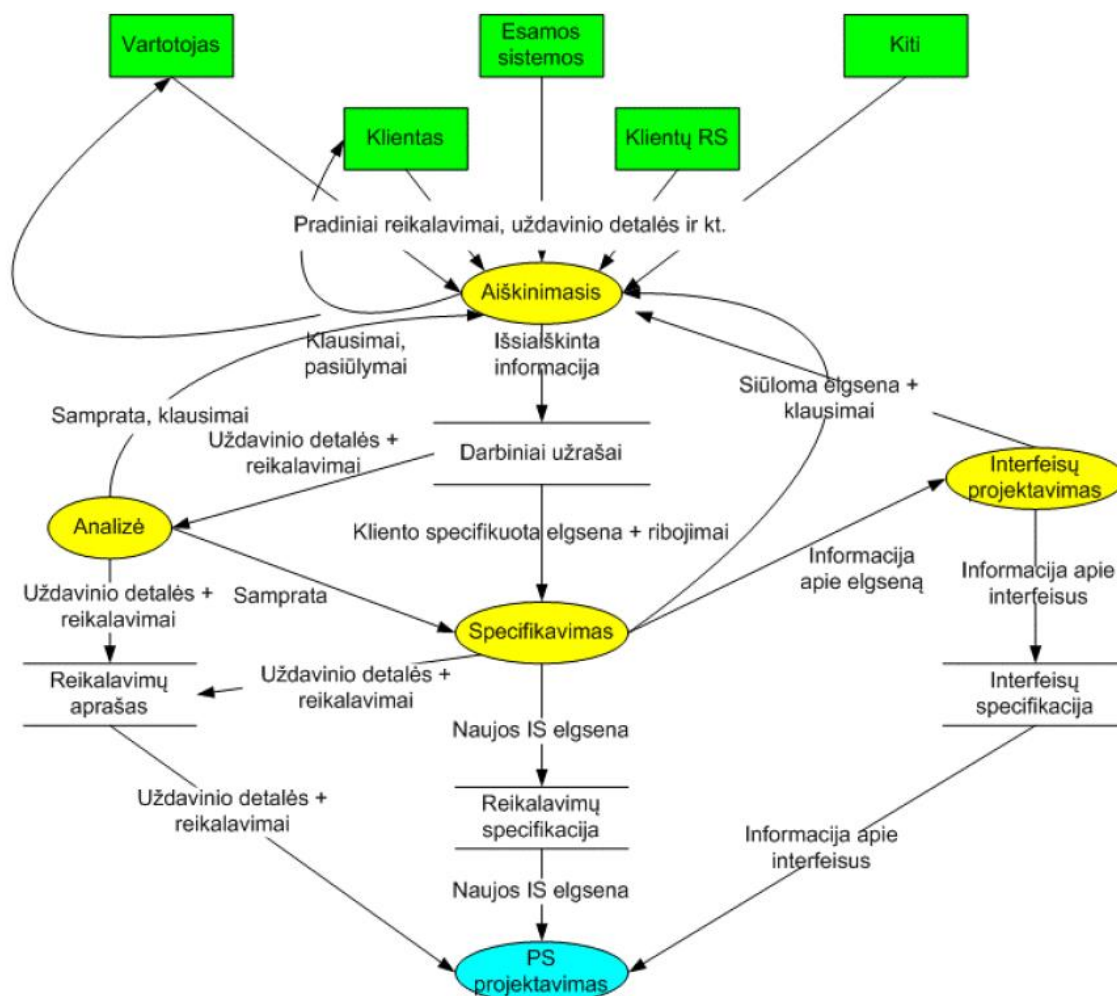
Išnagrinėjus šiame poskyriuje pateiktus pavyzdžius, galime teigti, kad kiberfizinėms-socialinėms sistemoms yra būdingos ne tik kiberfizinių sistemų savybės, bet ir žemiau paminėtos:

- Žmogaus elgsenos analizavimas ir poreikių nustatymas;
- Fizinio pasaulio pritaikymas pagal žmogaus poreikius;
- Sistemą sudaro heterogeninės dalys (tiek programos, tiek fizinė aparatūra integruota su programine įranga);
- Tiek sistemos naudotojai, tiek jos komponentai turi tą patį statusą, tačiau skirtingas teises (pavyzdžiui, apie lauko temperatūrą gali paklausti tiek žmogus, tiek komponentas).

## 2. Reikalavimų analizė ir modeliavimas

Šiame skyriuje aprašomi reikalavimų analizės veiklos pradiniai duomenys ir jos rezultatai bei tiriamos kompiuterinių sistemų kūrime naudojamos specifikavimo metodikos.

Remiantis [Bra02] procesų modeliu (žr. 4 pav.), analizės veikla apima verslo sistemos išsiaiškinimą, jos savybių ir problemų aprašymą. Šios analizės metu yra nagrinėjama ir modeliuojama verslo sistema, o ne kuriamoji programų sistema. Todėl analizės tikslas yra suprasti verslo sistemos pobūdį ir problemų priežastis.



4 pav. I. K. Bray proceso modelis [Bra02]

Analizės pradiniai duomenys yra iš dalykinės srities šaltinių (vartotojo, kliento ir t.t.) gauta informacija, o jos rezultatai yra:

1. Konceptinis verslo sistemos modelis;
2. Reikalavimų aprašymas.

Reikalavimų aprašymas yra labai detalus. Tai yra informacijos iš įvairių šaltinių, pateiktų įvairiomis formomis, sutvarkymas. Tokiame aprašyme turi būti:

1. Kuriamos programų sistemos poveikis verslo sistemai, kad būtų išspręstos analizės metu išryškintos verslo problemos. Šie aprašai turi pateikti visą reikiamą informaciją, bet kartu būti ir pakankamai abstraktūs;

2. Dalykinės srities modelis;
3. Sistemos ribų apibrėžimas;
4. Kuriamos sistemos bendravimo taisyklės su dalykine sritimi;
5. Preliminarus sistemos suskaidymas į komponentus.

Autoriai straipsnyje [SCC17] pateikia savo versiją apie tai, kodėl klasikiniai specifikavimo metodai netinka kiberfizinėms-socialinėms sistemoms. Tyrėjai nurodo, kad didžioji dalis metodikų remiasi tuo, kad veikimo aplinka yra sumodeliuojama reikalavimų surinkimo fazėje ir vėliau yra nekeičiama. Tokiu atveju, aplinkai pasikeitus, turi būti keičiama dalis modelio arba rečiau – visas modelis, gali keistis ir pasiūlytas sprendimas. Kiberfizinėms-socialinėms sistemoms klasikiniai specifikavimo metodai netinka, nes jos neturi pastovios aplinkos, nors privalo nuolat prie jos prisitaikyti.

Taip pat minėtame šaltinyje pateikiama išvada, jog kol kas nėra būdų, kaip nurodyti kokius modulius ar sistemos bendrauja su kitais moduliais arba kaip programinė įranga bendrauja su naudotojais ar aplinka. Dėl to atsiranda prisitaikančios elgsenos poreikis. Tai reiškia, kad KFSS veikimo metu reikalavimai nėra pastovūs, o kintantys. Kintančių reikalavimų analizavimui reikalingos naujos metodikos. Autorių nuomone, tokių metodikų atsiradimui svarbu pakeisti mąstymą. Būtent dėl šios priežasties tyrėjai, remdamiesi Baresi ir kitų autorių nuomone, atkreipia dėmesį į tai, kad ateityje kūrimo metu nebeliks skirtumo tarp sąvokų „kūrimo metu“ (angl. development-time) ir „veikimo metu“ (angl. runtime). Būtent toks susijungimas, autorių įsitikinimu, atvers naujas galimybes metodikoms, kurios revoliucionizuos kiberfizinė-socialinių sistemų kūrimą. Kaip teigia tyrėjai, tokioje situacijoje reikia specifiuoti ne programinę įrangą, o sistemą, kuri susideda iš programinės įrangos, analitikų, projektuotojų, naudotojų ir aplinkos.

Klasikinės specifikavimo metodikos programų sistemų reikalavimų inžinerijoje laikomos šios: struktūrinė, objektinė, paslaugų ir probleminė domeno analizė.

Struktūrinės analizės metodika pasižymi tuo, kad išplėtė reikalavimų inžinerijoje naudojamas kalbas – buvo įvesti koncepciniai modeliai poreikių ir reikalavimų specifikacijose [Bra02]. Tačiau ši metodika daro prielaidą, kad egzistuojanti sistema tenkina sistemos reikalavimus ir naujai kuriama sistema yra išvedama iš senosios. Taip pat paveldimas sistemos pagrindą sudarantis struktūrinis modelis.

Objektinė analizė faktiškai yra taikoma sistemai modeliuoti. I. Jacobson atkreipė dėmesį, kad objektinė inžinerija gali būti pradedama naudoti nuo to laiko, kai sistemos specifikacija yra sukurta [Jac96]. Tam tikros objekcinio modeliavimo technikos gali prisidėti prie dalykinės srities ir jos subdomenų modeliavimo, tačiau objekcinio aspekto įnašas privalumų nesuteikia [Bra02].

Dalykinės srities dekomponavimas į paslaugas ir paslaugomis grindžiamas modeliavimas tradicinę objektinę analizę papildoma naujomis veiklomis ir artefaktais [Ars04]. Šios paradigmos

naudojimas leidžia atsižvelgti į tai, kad jau pradinėje stadijoje yra skirtingus domenus atstovaujančios sistemos dalys. Be to, identifikuojant paslaugas turi būti nagrinėjami ir socialinės sistemos tikslai.

Taikant probleminę domeno analizę išreikštiniu būdu atskiriama tiriamoji dalykinė sritis ir kuriama sistema, taip pat pateikiami problemų aprašas dalykinėje srityje ir reikalavimai, kurie turi būti tenkinami [Jac01]. Vienos konferencijos pranešimo metu M. Jackson atkreipė dėmesį, kad ši paradigma gali būti taikoma kuriant kiberfizinės sistemas, tačiau išsamios analizės ir metodinės rekomendacijos nepateikė. Taigi, klausimas, ar tokio tipo analizė gali būti pakankama kiberfizinė-socialinių sistemų inžinerijoje, lieka atviras.

Taigi, analizuojant ir modeliuojant KFSS svarbu nustatyti:

1. Sprendžiamas problemas;
2. Reikalavimus;
3. Dalykinę sritį;
4. Sistemos bendravimą su dalykine sritimi, apimant ir naudotojus.

## **2.1. Sprendiniu grindžiami reikalavimai**

Knygoje „Requirements Engineering“ [Poh10] Kohl Pohl aprašo sprendiniu grindžiamus (angl. Solution-based) reikalavimus. Autorius teigia, kad sistemose, kuriose yra labai daug programinės įrangos, reikalavimai turi aprašomi taip, kad palengvintų minimos sistemos kūrimą. Su tam tikromis kūrimo metodikomis galima iš detalių ir formalių reikalavimų specifikacijos sugeneruoti dalį reikiamo kodo. Dėl tokio reikalavimų aprašymo poreikio 1980-tais metais buvo aprašytos trys pagrindinės perspektyvos, kuriomis naudojamosi iki šiol:

1. Duomenų perspektyva. Jos tikslas yra sutelkti visą dėmesį į duomenų, kurie bus naudojami ir valdomi sistemoje, aprašymą. Tiksliau, nekintantys duomenų aspektai yra išreiškiami per esybes arba sąryšius tarp esybių, atributų ir atributų tipų. Perspektyva aprašo reikalavimus duomenims ir duomenų struktūrą sistemoje, dėl to į darbą su duomenimis nėra atsižvelgiama. Dėl šios priežasties pasirinkus formalią reikalavimų specifikavimo kalbą ir duomenis aprašius pakankamai detaliam, duomenų struktūrą (arba ją atitinkantį kodą) galima sugeneruoti automatiškai. Pavyzdžiui, esybių-ryšių modeliai yra dažnai naudojami aprašant duomenis duomenų bazėse. Jeigu šie modeliai yra užtektinai detalūs, automatiškai galima sugeneruoti reliacinės duomenų bazės schemą.
2. Funkcinė perspektyva. Šios perspektyvos dėka yra aprašomi: veiksmi (funkcijos), kuriuos turės atlikti sistema, manipuliavimas duomenimis kiekviename veiksmo ir veiksmų įvestis-išvestis. Tradiciškai, funkcinėi perspektyvai aprašyti yra naudojama duomenų srautų diagrama. Kaip ir duomenų perspektyvos atveju, pakankamai detalūs reikalavimai iš funkcinė reikalavimų specifikacijos leidžia sugeneruoti visą arba bent dalį reikalingo kodo.

3. Elgsenos perspektyva. Jos tikslas yra aprašyti bendrą sistemos elgseną. Šioje perspektyvoje yra pateikiami išorės daromas poveikis sistemai ir sistemos reakcija į jį. Taip pat apibūdinamas ir ryšys tarp išorinio poveikio ir sistemos reakcijos, pateikiamos galimos sistemos būsenos (arba režimai) bei galimi perėjimai tarp būsenų. Dėl to dažnai naudojamos būsenų diagramos. Kaip ir aukščiau paminėtose perspektyvose, pakankamai detaliam aprašyti reikalavimai leidžia sugeneruoti bent dalį kodo arba sistemos elementus.

Nors pateiktos perspektyvos sutelkia dėmesį į skirtingus kuriamo sprendimo aspektus, jos aprašo tą patį domene nagrinėjamą objektą. Visos trys perspektyvos yra glaudžiai susijusios:

1. Sistemos duomenų specifikavimas duomenų ir funkcinėje perspektyvose. Duomenų perspektyva aprašo reikalavimus duomenims, pavyzdžiui, naudojant esybių-ryšių diagramas. Tuo tarpu funkcijose šie duomenys yra naudojami kaip įėjimas ir išėjimas. Pavyzdžiui, duomenų srautų diagramoje pateikiama, kaip duomenys keliauja tarp procesų ir duomenų saugyklų. Tačiau funkcinė perspektyva nenagrinėja ryšių tarp esybių (pavyzdžiui, ar gali būti tik viena esybė, ar leistinos kelios).
2. Sistemos funkcijų specifikavimas funkcinėje ir elgsenos perspektyvose. Funkcinė perspektyva aprašo sistemos suteikiamas funkcijas, o elgsenos – kada ir kokia funkcija turi būti iškviečiama. Elgsenos perspektyvoje papildomai nurodoma sistemos būsena prieš ir po funkcijos atlikimo.

Apibendrinant, ši metodika nėra užtektina kiberfizinė-socialinių sistemų reikalavimų analizei atlikti, nes tokios sistemos pilnai arba dalinai susideda iš autonomiškų posistemių. Joms specifikuoti reikalinga informacija apie tai, koks yra jų siekiamas tikslas, tačiau dėl jų dinamiško ir nenuspėjamo tokių posistemių elgesio aprašyti visus išorinius veiksnius nėra įmanoma, o tai yra vienas iš elgsenos perspektyvos reikalavimų. Taip pat šios metodikos pritaikymo atveju nėra aiškus sistemos bendravimas su dalykine sritimi ar naudotojais.

## **2.2. Agentais grindžiamas specifikavimas**

A. Lupeikienė knygoje „Teoriniai ir technologiniai informacinių sistemų aspektai“ [Lup07] nurodo, kad terminas „agentas“ neturi vienos apibrėžties, todėl autorė pateikia apibendrintą šios sąvokos suvokimo variantą. Jos teigimu, agentas, tai „esybė, suvokianti aplinką savo jutikliais ir daranti poveikį tai aplinkai savo manipulatoriais“. Taip pat autorė knygoje aprašo agentines technologijas, kurios yra skirtos specifikuoti sistemas su nuolatos kintančia aplinka. Būtent tokios yra ir kiberfizinės-socialinės sistemos.

Šiame šaltinyje teigiama, kad agentas yra autonomiškas, jeigu jis funkcionuoja be nurodymų iš aplinkos. Agentui reikia valdyti turimus išteklius ir turėti įgaliojimus norimiems veiksams atlikti. Tam specifikacijoje turi būti aprašomi sistemai priklausantys ištekliai, jų kiekiai, galimos

būsenos ir ryšiai. Taip pat turi būti paminėti galimi sistemos veiksmai, reikalingos jos būsenos tiems veiksams atlikti bei jų įgaliojimai (ribojimai).

Agentas taip pat yra ir reaguojantis, jeigu jis sugeba prisitaikyti prie kintančios aplinkos. Dėl šios priežasties, specifikacijoje turi atsirasti agento turimi jutikliai, jų aprašai (interfeisai, nefunkciniai reikalavimai), galimos sistemos būsenos ir aprašymas, kaip jos turi pasikeisti priklausomai nuo iš jutiklių gaunamos informacijos. Būsenų pasikeitimas gali būti paprastas (pavyzdžiui, pasiekus tam tikrą temperatūrą įranga turi išsijungti) arba sudėtingas, priklausantis nuo buvusios būsenos ar kitokių veiksmų (pavyzdžiui, jeigu dabar yra saulėta, kambaryje nėra žmonių ir kambario temperatūra viršija 26 laipsnius, ir norint, kad kambario temperatūra nebekiltų, reikia nuleisti žaliuzes). Tam, kad įvykdytų šiuos veiksmus ir priimtų atitinkamus sprendimus, sistema privalo turėti aprašytus tikslus (pastarajame pavyzdyje pateiktas tikslas – taupyti energiją, sumažinant vėsinimui reikalingos energijos kiekį).

Samprotaujantys agentai sugeba priimti sprendimus, kurie iš anksto nebuvo specifikuoti. Kitaip tokie agentai dar yra suprantami kaip dirbtinis intelektas. Norint turėti šiuos agentus, kas yra būtina KFSS, reikia specifikuoti turimas žinias apie dalykinę sritį ir kiekvieno agento tikslus.

Kadangi visą sistemą gali sudaryti daugiau nei vienas agentas, specifikacijoje taip pat reikia nurodyti tarp jų esančius ryšius ir agentų tarpusavio sąveiką. Jie, keisdami informaciją ir vykdydami savo užduotis, tampa socialinėmis esybėmis.

Iš pirmo žvilgsnio gali atrodyti, kad specifikacijoje atsiranda viskas, ko reikia kibernetinėms-socialinėms sistemoms specifikuoti, nes specifikacija apima visus galimus pasaulius. Tačiau ši metodika taip pat nėra užtektina, nes ji yra skirta programinei įrangai kurti, dėl to jos pritaikymo metu nėra išskiriami fizinis ir kibernetinis pasauliai. Pavyzdžiui, specifikuojant jutiklius nėra atsižvelgiama į kibernetinius ir fizinius aspektus. Dėl to užtikrinti fizinio pasaulio pritaikymą pagal žmogaus poreikius tampa sudėtinga, nes pasikeitęs poreikis gali būti ne funkcionalume (kibernetiniame pasaulyje), bet fiziniame pasaulyje. Žmogui įsigijus didesnę išmanųjį šaldytuvą, jo funkcionalumas išlieka toks pats, tačiau fiziniame pasaulyje atsiranda didesnė šaldymo erdvė, kurios temperatūrą reikia palaikyti pagal aprašytus nefunkcinius reikalavimus. O didesnė šaldymo erdvė lemia tai, kad reikalingi galingesni kompresoriai, kurie turi atitikti jiems aprašytus nefunkcinius reikalavimus.

### **2.3. Probleminio domeno analizė**

Michael Jackson pristatė probleminius freimus (angl. Problem Frames), kurių funkcija yra atlikti kuriamos sistemos probleminę domeno analizę. Literatūriniuose šaltiniuose [Jac01], [WTN06] ir [Lup07] yra pateikiami šie probleminių freimų šablonai:

1. Taisyklėmis valdoma sistema (angl. Required Behavior). Šis probleminis freimas naudojamas, kai bent dalis fizinio pasaulio elgesio turi būti kontroliuojama norint atitikti jam keliamus reikalavimus. Problema yra sistemos, kuri kontroliuos minimą elgesį, sukūrimas;
2. Komandomis valdoma sistema (angl. Commanded Behavior). Freimas naudojamas, kai bent dalis fizinio pasaulio elgesio turi būti kontroliuojama pagal operatoriaus siunčiamas komandas. Problema yra sistema, kuri priims operatoriaus komandas ir perteiks atitinkamą kontrolę fiziniam pasauliui, sukūrimas;
3. Sistema-įrankis (angl. Simple Workpieces). Šis probleminis freimas naudojamas, kai reikalingas įrankis, kuris leistų naudotojui sukurti ir koreguoti įvairius kompiuterinius tekstus ar grafinius objektus (ar panašius dalykus), kuriuos galima būtų panaudoti įvairiausiais būdais (pavyzdžiui, kopijavimui, spausdinimui ar analizavimui). Problema yra sukurti sistemą, kuri elgtųsi kaip paminėtas įrankis;
4. Diskretaus informavimo sistema (angl. Information Display). Freimas naudojamas, kai yra reikalinga informacija apie fizinio pasaulio (viso arba dalies) būseną ir/arba elgesį. Problema yra sukurti sistemą, kuri rinktų šią informaciją, ją atitinkamai apipavidalintų ir perduotų atitinkamiems procesams;
5. Transformavimo sistema. Šis probleminis freimas naudojamas, kai yra gaunami duomenys, kurie turi būti transformuoti pagal atitinkamus kriterijus. Problema yra sukurti sistemą, kuri sukurtų reikiamą rezultatą iš gautų duomenų.

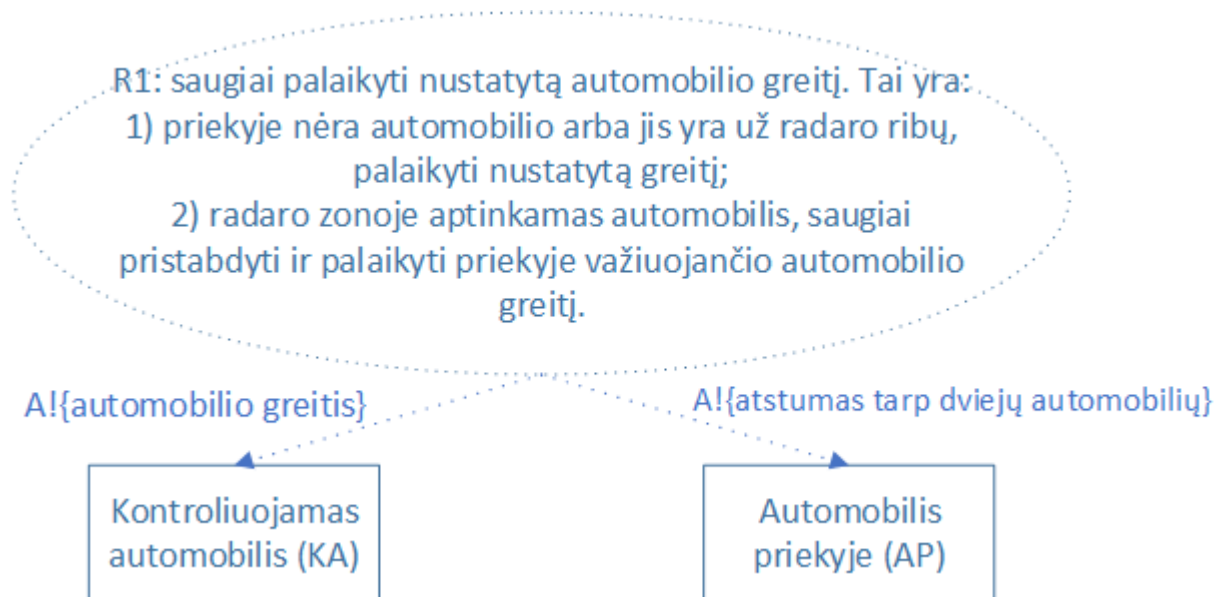
Visi aukščiau paminėti probleminiai freimai padeda atrasti, surinkti reikiamus reikalavimus ir modeliuoti visą sistemą. Konkretaus freimo pasirinkimas priklauso nuo sistemos pobūdžio, tačiau gali būti, kad sistema yra sudaryta ir iš kelių probleminių freimų.

Nagrinėdami probleminius freimus, toliau šiame poskyriuje rėsimės [LZZ13] straipsnio medžiaga, kurioje yra aprašoma kiberfizinėms sistemoms skirta metodika, kuri yra paremta problemineis freimais.

Visi pristatyti freimai remiasi tuo, kad naudotojo poreikiai yra realiame pasaulyje ir išreiškia norimus domeno reiškinius (elgesį), kurie gali būti visiškai nesusiję su programine įranga. Norint atitikti šiuos poreikius, programinė įranga paveikia realų fizinį pasaulį per bendrus reiškinius (angl. shared phenomena). Tokie poveikiai dažniausiai keliauja per kelis realiam pasaulyje esančius domenų, kol galiausiai yra išpildomas norimas reiškinys.

Specifikacija prasideda nuo naudotojų poreikių identifikavimo. Jie turėtų būti išreikšti per reiškinius fiziniame pasaulyje. Poreikiai gali būti užrašyti tiek tekstu, tiek diagrama, tačiau dėl kompaktiškumo ir aiškumo rekomenduojama rinktis antrąjį užrašymo būdą. Pateiktame diagramos pavyzdyje (žr. 5 pav.) fiziniame pasaulyje yra išskiriami du reiškiniai: automobilio greitis ir atstumas tarp dviejų automobilių.





5 pav. Naudotojų poreikio pavyzdys [LZZ13]

Toliau specifikuojant yra įvertinama naudotojų poreikio įgyvendinimo galimybė. Kiberfizinės sistemos stebi ir kontroliuoja fizinius domenų. Reikalavimuose yra tikrinama ar reiškiniai, kuriuos reikės stebėti, gali būti stebimi ir atitinkamai ar tie, kuriuos reikės kontroliuoti, gali būti kontroliuojami. Jeigu bent vienas reiškinys neatitinka reikalavimo, tuomet naudotojų reikalavimas turi būti tikslinamas. Pateiktame pavyzdyje automobilio greitis turi būti kontroliuojamas ir jis gali būti kontroliuojamas, o atstumas iki automobilio esančio priekyje turi būti stebimas ir jis gali būti stebimas. Tai nurodo, kad reikalavimas yra įgyvendinamas.

Atlikus šiuos specifikacijos veiksmus reikia transformuoti naudotojų problemas ir patikrinti įrangos atitikimą gautiems reikalavimams. Būtent šiam žingsniui atlikti padeda probleminiai freimai.

Probleminis freimas išskiria tris dalykus:

1. Naudotojų poreikiai (R);
2. Domenų prielaidos (P). Aprašomos realiame pasaulyje esančių domenų savybės;
3. Įrangos specifikacija (S). Pateikia galimus įvykius bendruose reiškiniuose.

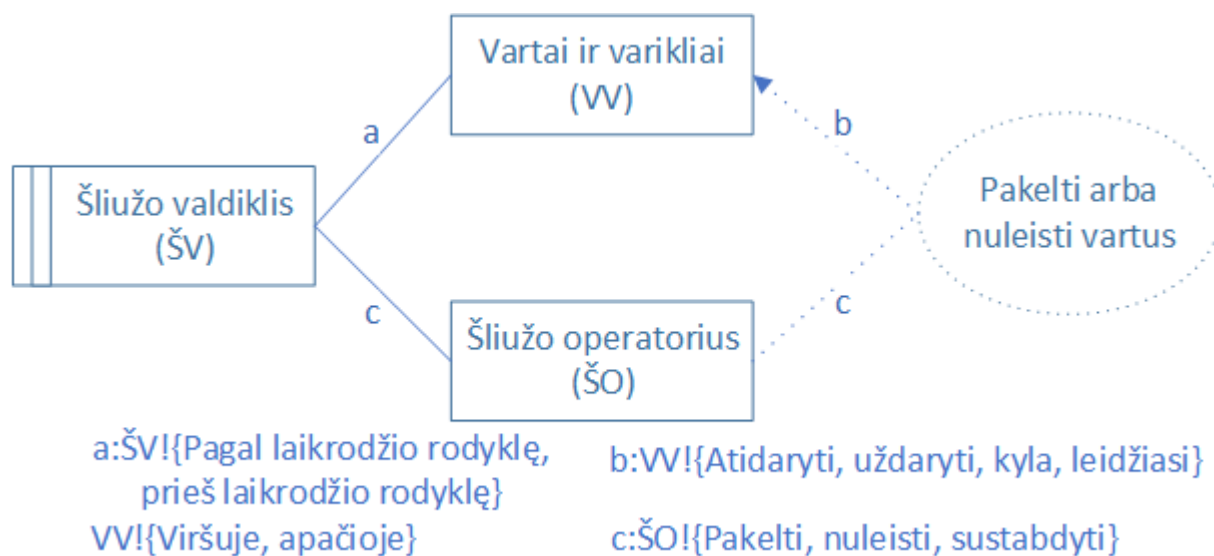
P, S | R

1 formulė. Ryšys tarp poreikių (R), domenų prielaidos (P) ir specifikacijos (S)

1 formulėje yra parodytas pavyzdys, kaip gali būti aprašomas ryšys tarp R, P ir S. Jeigu įranga, atitinkanti pateiktą specifikaciją S, yra įrašyta realiame pasaulyje kuriame domenų savybės yra P, tai naudotojų poreikiai R bus patenkinti.

Probleminiuose freimuose paprasti stačiakampiai žymi domenų, esančius realiame pasaulyje; stačiakampiai su brūkšneliais – programinę įrangą; elipsės nurodo loginį ryšį tarp esybių, kas sutapatinama su reikalavimu (arba kitaip sakant – norimu reiškiniu); vientisos linijos reiškia, kad

jomis jungiamos esybės gali dalintis įvairiais elementais (pavyzdžiui, būsenomis ar resursais); brūkšninės linijos yra nuorodos į reikalavimus (žr. 6 pav.).



6 pav. Komandomis valdomos sistemos pavyzdys [LZZ13]

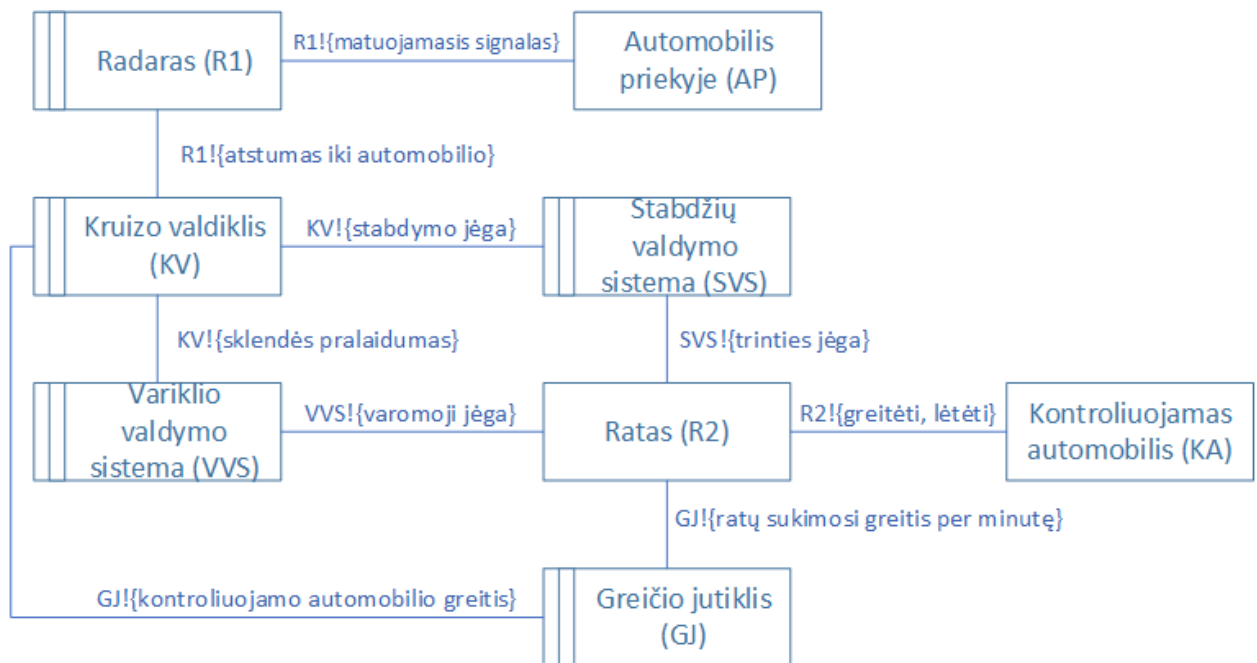
Apibūdinant įrangos problemą kartu yra aprašoma ir reikalavimų specifikacija. Šiam veiksmui atlikti yra įvedamas „reikalavimų transformavimo“ terminas. Patys pirmieji reikalavimai, kurie atstovauja pradinę problemą, yra gaunami iš naudotojų poreikių. Tuomet pradinė problema yra transformuojama į realaus pasaulio domenų problemas. Galiausiai šių domenų problemas transformuojamos į programinės įrangos problemas, kurios yra paremtos tam tikromis domenų savybėmis (pavyzdžiui, vartai gali būti pakeliami arba nuleidžiami sukant variklį prieš arba pagal laikrodžio rodyklę).

Straipsnyje siūloma specifikuojant naudoti du modelius:

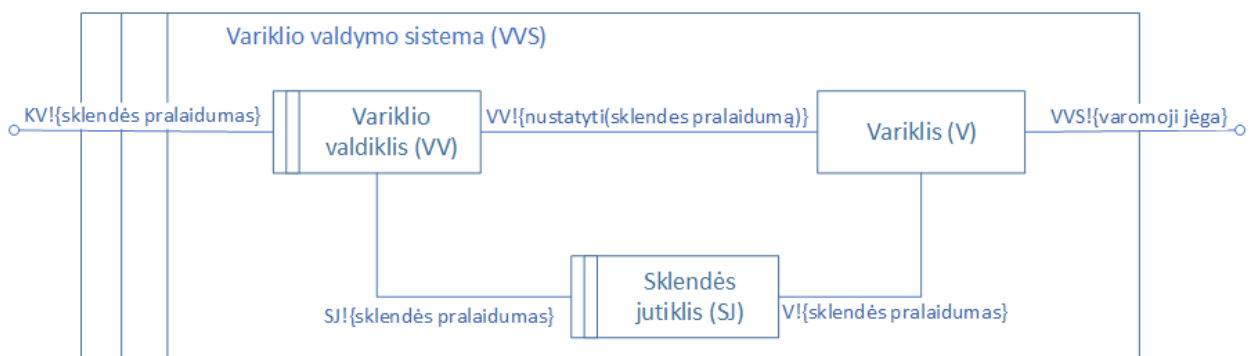
1. Struktūros modelį (probleminį freimą);
2. Tikslų modelį.

Struktūros modelis susideda iš domenų (tiek fizinių, tiek programinės įrangos) ir bendrų reiškinių. Fiziniai domenai gali būti sudaryti tik iš kitų fizinių domenų, tuo tarpu programinės įrangos domenai gali būti sudaryti tiek iš fizinių, tiek ir iš kitų programinės įrangos domenų. Domenai tarpusavyje yra sujungti bendrais reiškiniiais.

Autoriai iliustruoja savo teiginius pasirinkdami kruizo sistemos, kuri yra kiberfizinių sistemų poklasis, pavyzdį. Pateiktame struktūros modelio pavyzdyje (žr. 7 pav.) fiziniai domenai yra šie: automobilis priekyje, ratas, kontroliuojamas automobilis. Tuo tarpu programinės įrangos domenai yra: radaras, kruizo valdiklis, stabdžių valdymo sistema, variklio valdymo sistema, greičio jutiklis. Iš jų keli programinės įrangos domenai yra kompleksiniai, t.y. sudaryti iš kitų domenų junginio. Pavyzdžiui, variklio valdymo sistema (žr. 8 pav.) susideda iš variklio, kuris yra fizinis domenas ir iš programinės įrangos: variklio valdiklio ir sklendės (angl. throttle) jutiklio.



7 pav. Kruizo sistemos struktūros modelis [LZZ13]



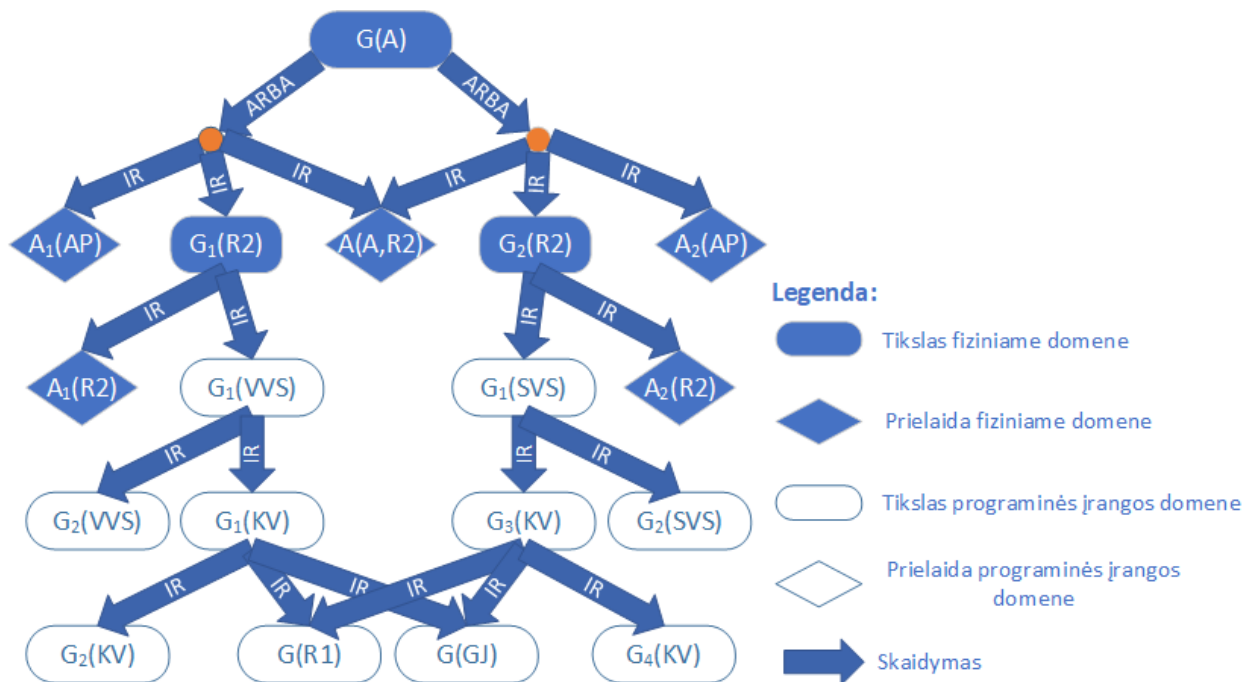
8 pav. Variklio valdymo sistemos struktūros modelis [LZZ13]

Struktūros modelio privalumas yra tas, jog jis parodo iš kokių domenų susideda nagrinėjama kiberfizinė sistema. Tačiau šis modelis turi ir trūkumą – jame neatsispindi ką tie domenai atlieka. Taip pat nėra žinoma, ar domenai tarpusavyje bendrauja nenutraukiamai.

Atsakyti į šiuos klausimus padeda tikslų modelis. Jo dėka reikalavimai yra aiškiai priskiriami atitinkamiems domenams. Šis modelis ypatingas ir tuo, kad jis leidžia patikrinti struktūros modelio teisingumą.

Tikslų modelis susideda iš: tikslų, pavyzdžiui,  $G(X)$ , kur  $X$  yra konkretus domenas; domeno prielaidos, pavyzdžiui,  $A(Y)$ , kur  $Y$  yra konkretus domenas; sąryšių tarp  $T$  ir  $P$ . Vienas iš tikslų modelio pavyzdžių būtų modelis pritaikytas anksčiau minėtai kruizo kontrolės sistemai (žr. 9 pav.). Pati viršūnė  $G(A)$  reiškia norimą pasiekti tikslą – palaikyti saugų greitį kontroliuojamame automobilyje. Šis tikslas gali būti įgyvendintas dviem būdais (ryšys „arba“). Kad būtų aiškiau, pateiksime tik vieną iš dviejų variantų, iš kairės pusės. Norint pasiekti  $G(V)$  keliamą tikslą, reikia atitikti visus tris elementus:

1.  $A_1(AP)$  – prielaida, kad priekyje nėra automobilio arba jis yra ne radaro pasiekiamoje (saugioje) zonoje;
2.  $A(A,R2)$  – prielaida, kad rato greitis yra lygus kontroliuojamo automobilio greičiui.
3.  $G_1(R2)$  – tikslas, kad ratas suksis kruizo sistemoje nustatytu greičiu. Šis tikslas yra labai abstraktus, dėl to jis gali būti skaidomas į smulkesnes dalis:
  - a.  $A_1(R2)$  – prielaida, kad ratui suteikus tinkamą jėgą jis suksis atitinkamu greičiu;
  - b.  $G_1(VVS)$  – tikslas, kad variklio valdymo sistema apskaičiuos tinkamą jėgą, kuri leis pasiekti kruizo greitį. Šis tikslas sudarytas iš dar kelių dalių:
    - i.  $G_2(VVS)$  – tikslas, kad variklio valdymo sistema apskaičiuos tinkamą jėgą pagal sklendės pralaidumą;
    - ii.  $G_1(KV)$  – kruizo valdiklis apskaičiuoja reikiamą sklendės pralaidumą. Šis tikslas sudarytas iš:
      - 1)  $G_2(KV)$  – kruizo valdiklis apskaičiuoja reikiamą sklendės pralaidumą pagal atstumą iki priekyje esančio automobilio ir dabartinį kontroliuojamo automobilio greitį;
      - 2)  $G(R1)$  – radaras atiduoda atstumą iki priekyje esančio automobilio;
      - 3)  $G(GJ)$  – greičio jutiklis atiduoda dabartinį kontroliuojamo automobilio greitį.



9 pav. Kruizo kontrolės sistemos tikslų modelis [LZZ13]

Toks tikslo išskaidymas atitinka anksčiau minėtą problemos transformacijos procesą. Sudaryto medžio schemas viršuje yra vaizduojama fiziniame pasaulyje iškilusi naudotojo problema. Tuo tarpu medžio lapai iliustruoja išgrynintas problemas arba prielaidas, dažniausiai perkeltas į programinės įrangos pasaulį, kurias įgyvendinus bus išspręsta naudotojo problema. Šiuo atveju, įgyvendinus  $G_2(VVS)$ ,  $G_2(KV)$ ,  $G(R1)$ ,  $G(GJ)$ ,  $A_1(R2)$  ir  $A_1(AP)$  automobilio greitis bus lygus kruizo sistemoje nustatytam greičiui. Verta paminėti, kad keičiant struktūros modelį ar

turint juos kelis, galima detalizuoti ir tikslų modelio gautus lapus (pavyzdžiui,  $G_2(VVS)$  gali būtų smulkinamas), tačiau viskas priklauso nuo to, kokio detalumo reikia.

Probleminių freimų dėka yra aprašomi visi kiberfizinių sistemų pasauliai ir jų tarpusavio ryšiai. Tačiau atsakymo į klausimą, ar galima specifikuoti naudojantis šia metodika, kai atsiranda papildomas socialinis pasaulis, nėra.

## 2.4. Kiti specifیکavimo būdai

Straipsnyje [DLV11] teigiama, kad modeliu paremtos metodikos turi privalumų. Pavyzdžiui, vienas iš jų yra tas, kad modeliai yra deterministiniai. Formaliai aprašius modelį, mes jį galime vertinti, o tokia galimybė reiškia, kad modeliui pateikus tą pačią įvestį, jis visuomet atiduos tą patį rezultatą. Pavyzdžiui, šaltinyje yra pateikiama metodika „Ptolemy II“, kuri naudoja į aktorius orientuotą metodiką, kad modeliuoti įvairius komponentus. Aktoriai gali bendrauti su kitais aktoriais. Bendravimo taisyklės yra aprašytos skaičiavimo modelyje (angl. model of computation), kuris yra įgyvendinamas vadovo. Vadovas gali būti komponentas iš modelio arba sub-modelio. Šie vadovai gali būti sujungti į vienetinį hierarchinį modelį. Regionas, kurį aprašo pasirinktas vadovas yra vadinamas domenu. Taip sudaromas sistemos modelis.

Šaltiniuose [GYa13] ir [Zha13] specifیکavimui naudojama AADL (angl. Architecture Analysis and Design Language). Ši kalba skirta sistemos projektavimui, tačiau, autorių nuomone, šiek tiek ją pakoregavus galima naudoti ir specifیکavimui. Šių pakeitimų pagalba galima aprašyti sistemos elgesį, dinamines funkcijas ir erdvės-laiko (angl. spatial-temporal) reikalavimus.

Šaltinyje [RRW12] pristatoma „input/output automata“ metodika, kuri modeliuoja reikalavimus, įvairių komponentų elgesį jiems veikiant (angl. runtime). Šios metodikos pritaikymo metu yra modeliuojami komponentų įvesties ir išvesties reikalavimai.

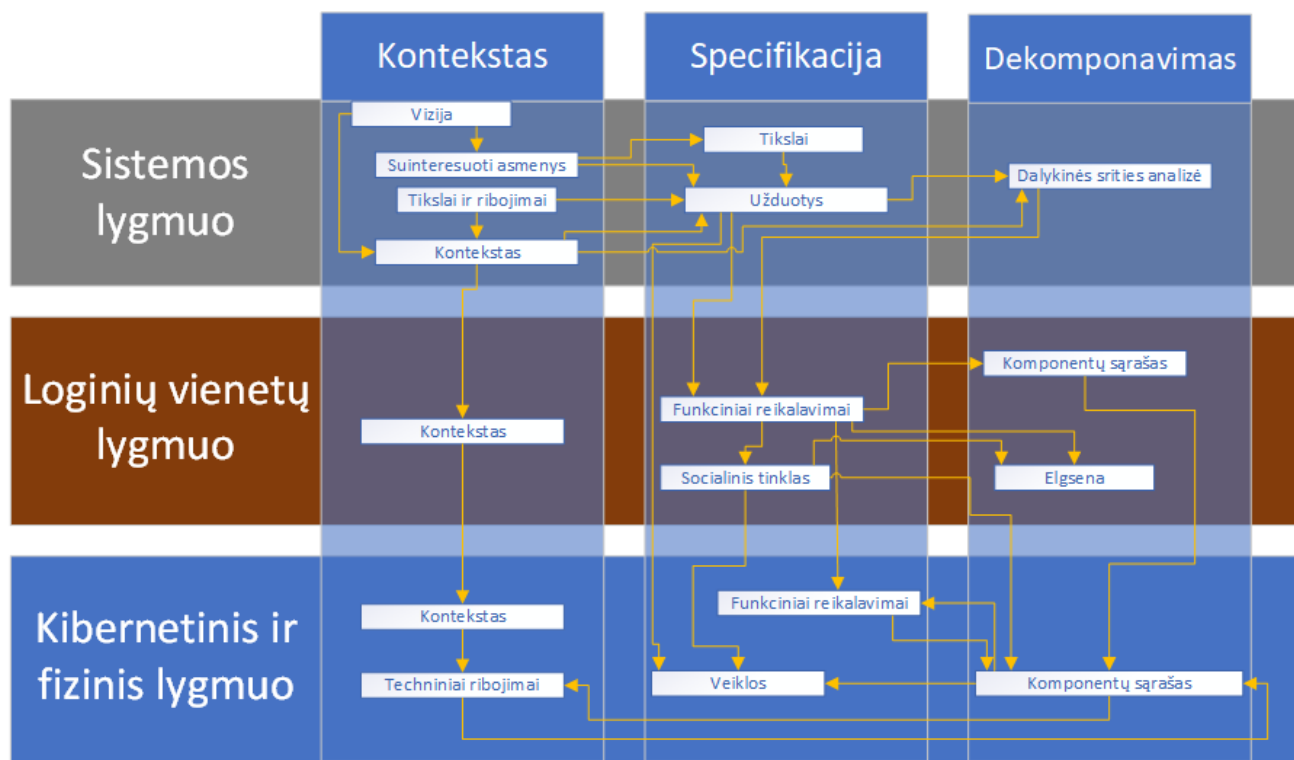
Džonas, Edvardas ir kiti autoriai straipsnyje [ELM+11] specifیکavimui siūlo naudoti „Programming Temporally Integrated Distributed Embedded Systems (PTIDES)“. Panašiai kaip minėtoje „I/O automata“ metodikoje, modelių pagalba yra aprašoma sąveika tarp įvairių komponentų, kurie vadinami agentais. Šios metodikos privalumas yra tas, kad specifikuotus komponentus galima simuliuoti, taip patikrinant ar reikalavimai ir specifیکacija yra teisingi. O tam tikrais atvejais galima sugeneruoti reikalingą kodą automatiškai.

Apibendrinant galima teigti, kad nėra vieno šablono, pagal kurį turėtų būti aprašomos KFSS. Įvairūs specifیکavimo būdai yra pritaikyti tik programinei įrangai kurti, neįtraukiant fizinio, o dažniausiai ir socialinio, pasaulių. Dėl šios priežasties nėra užtikrinamos reikiamos kiberfizinių-socialinių sistemų savybės.

### 3. Dalykinės srities analizė ir modeliavimas kuriant KFSS

Magistrinio darbo trečiame skyriuje pateikiamas reikalavimų metodinis karkasas ir reikalavimų pavyzdinis modelis. Kadangi kiberfizinėms-socialinėms sistemoms yra svarbu žinoti apie aplinką (išorinius domenų), todėl metodiniame karkase siūloma aprašant sistemą taikyti probleminio domeno analizę, tiksliau – naudoti probleminius freimūs, kuriems yra pasiūlyti pakeitimai, norint juos taikyti KFSS.

Nors analizės rezultatų nusakymui gali būti taikomi įvairūs metodiniai karkasai (pvz., J. A. Zachmano karkasas arba nefunkcinių reikalavimo nuleidimo žemyn metodas), tačiau jie nėra pilnai tam tinkami, nes tokie karkasai neatsižvelgia į tai, kad žemiausiuose lygmenyse yra kiberfizinės sistemos. Viršutiniuose sluoksniuose aprašomos savybės gali būti nepakankamos detalizuojant jas žemiausiuose lygmenyse. Dėl to šiame darbe siūlomas reikalavimų metodinis karkasas skirtas kiberfizinėms-socialinėms sistemoms (žr. 10 pav.), kuris yra sudarytas remiantis „Requirements Engineering and Management for software-intensive Embedded Systems (REMsES)“ [KMP+06] pagrindu. Originalus reikalavimų metodinis karkasas yra skirtas įterptinėms sistemoms, kurios yra vienas iš kiberfizinės-socialinės sistemos įgyvendinančių sudedamųjų dalių tipų.



10 pav. Reikalavimų metodinis karkasas skirtas KFSS

Reikalavimų metodiniame karkase (žr. 10 pav.) yra parodyta, iš kokių požiūrio taškų (vertikalūs lygmenys) nagrinėjama sistema, kokios ir koku detalumu turi būti aprašytos jos savybės. Šis metodinis karkasas yra sudarytas iš horizontalių ir vertikalųjų lygmenų. Žemiau pateikiami horizontalūs lygmenys:

1. **Sistemos lygmuo.** Aprašomas sistemos poveikis išorei;
2. **Loginių vienetų lygmuo.** Dominuojantys šio lygmens elementai yra sistemos loginės dalys. Apibūdinamos sudedamosios dalys ir vidiniai interfeisai. Aukštesniojo lygmens aprašai yra išskaidomi ir detalizuojami, tačiau neskirstomi į kibernetinius ir fizinius. Taip pat šiame lygmenyje į sistemos elementus yra žiūrima kaip į socialines esybes. Tai reiškia, kad turi būti užtikrinta galimybė keistis informaciniais vienetais ir užduotimis. Vienodomis teisėmis dalyvauja ir žmonės, ir kiberfiziniai komponentai. Pavyzdžiui, vienas iš sistemos tikslų yra „Prižiūrėti veją“ (susideda iš „Palaistyti veją“ ir „Nupjauti veją“). Namo lango jutiklis informuoja laistymo sistemą apie aukštą lauko temperatūrą. Tuomet laistymo sistema turi „Palaistyk veją“ tikslą, kurio pasiekimas susideda iš dviejų užduočių „Gauk saulės intensyvumą“ ir „Įjunk vejos laistytuvus“. Pirmosios užduoties laistymo sistema negali atlikti, todėl socialinio tinklo dėka yra paprašoma kitų komponentų pagalbos, kurie gali būti ir iš kitų KFS (kaimyno namo). Antroji užduotis yra vykdoma tik tuomet, kai yra gautas atsakas „Intensyvi saulė“ iš pirmąją užduotį atlikusio komponento;
3. **Kibernetinis ir fizinis lygmuo.** Šiame lygmenyje loginiai vienetai yra tipizuojami, parenkama jų topologija (jungčių, konfigūracijų, dalių tarpusavio ryšio visumos schema). Nagrinėjamų sistemų atveju skiriamos kibernetinės ir fizinės sudedamosios dalys, kurias apima jutiklius ir manipulatorius.

Kiberfizinių-socialinių sistemų reikalavimų metodinio karkaso schemoje pateikiami vertikalūs lygmenys yra:

1. **Kontekstas.** Šiame stulpelyje aprašai remiasi išoriniu pasauliu. Gali atsirasti įvairūs ribojimai (pvz., šalies įstatymai) arba naudojamos išorinės sistemos (pvz., taksi paslaugoms naudoti tik „Uber“ arba naudoti tik „Windows 10“ operacinę sistemą);
2. **Specifikacija.** Aprašomi sistemos reikalavimai atsižvelgiant į nuo konteksto priklausančius ribojimus. Gali atsirasti ir klientų keliami reikalavimų ribojimai (pvz., specifinė seka vykdant kokią nors užduotį, nepriklausomai nuo to, kad ji galėtų būti įvykdoma greičiau);
3. **Dekomponavimas.** Šiame lygmenyje pateikiamas sistemos dekompoziciją nusakantis aprašas. KFSS reikalavimų specifikacija turi apimti informaciją apie pagrindinius komponentus, nes vien tik aukšto lygmens sistemos specifikacijos nepakanka, kad iš jos būtų išvesta kiberfizinių komponentų specifikacija.

Diagramoje matomų esybių ir jų sąryšių paaiškinimai:

1. **Vizija.** Aprašo sistemos viziją.
2. **Suinteresuoti asmenys** (angl. stakeholders). Pateikiamas suinteresuotų asmenų aprašas. Įmonėje gali dirbti daug įvairių žmonių, bet dėl kuriamos sistemos suinteresuota gali būti tik jų dalis;

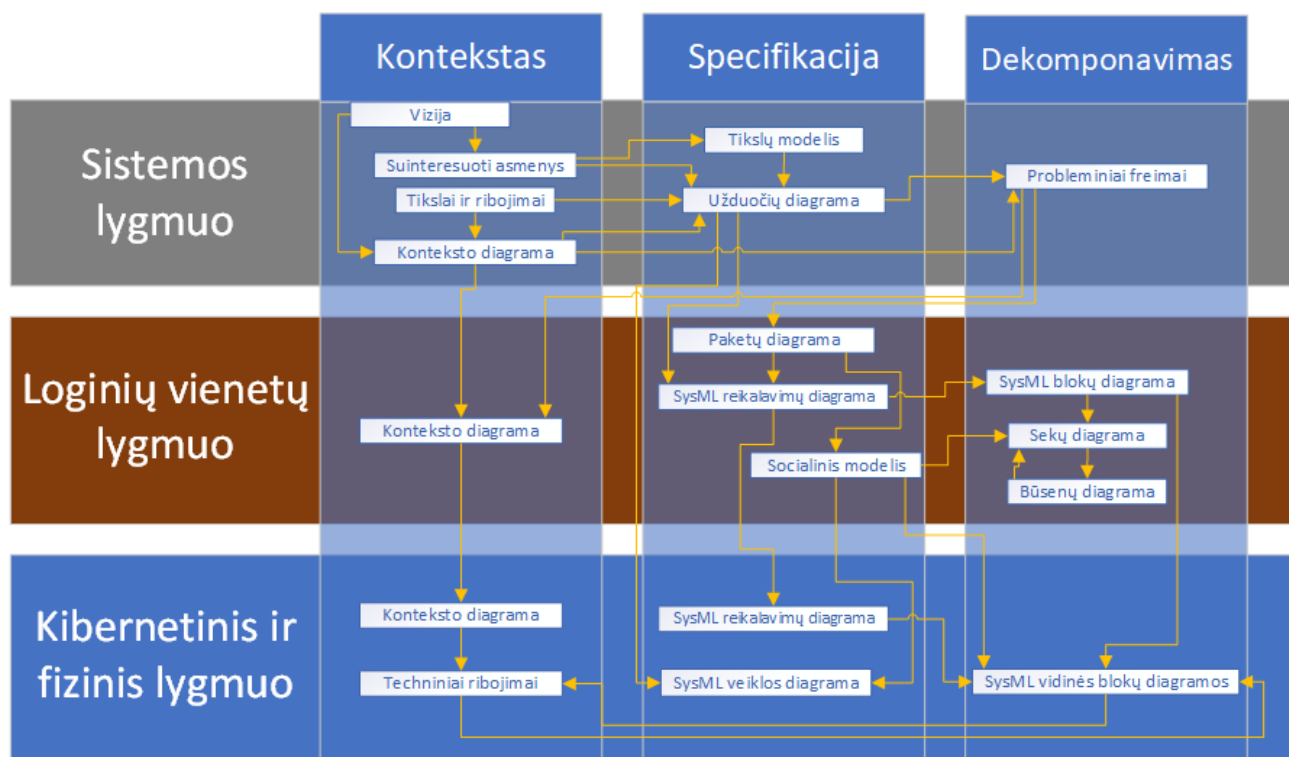
3. **Tikslai ir ribojimai.** Kuriamos sistemos tikslai, nefunkciniai reikalavimai ir jai keliami ribojimai (teisiniai ar pan.);
4. **Kontekstas.** Aprašo domenų, kurie turi įtakos kuriamai sistemai. Žemesniuose lygmenyse kuriama sistema išskaidoma atitinkamais komponentais;
5. **Tikslai sistemos lygmenyje.** Aprašomi tikslai, kuriuos turi įgyvendinti kuriama sistema;
6. **Užduotys** (angl. use case). Aprašomos užduotys, kurių įgyvendinimas reikalingas tikslams pasiekti;
7. **Dalykinės srities analizė.** Apibūdina, kokių pokyčių reikia analizuojamoje dalykinėje srityje, kad įgyvendinti užsibrėžtus tikslus;
8. **Funkciniai reikalavimai loginių vienetų lygmenyje.** Aprašomi loginiams vienetams reikalingi funkciniai reikalavimai;
9. **Socialinis tinklas.** Aprašo socialinį lygmenį tarp KFS ar jos narių. Žinant kokias veiklas turi atlikti komponentai, reikia aprašyti jų kooperavimąsi (protokolą, kaip apsikeičiama ir pasiskirstoma užduotimis) ir galimų užduočių sąrašą;
10. **Komponentų sąrašas loginių vienetų lygmenyje.** Sistema dekomponuojama į loginius vienetus, neišskiriant kibernetinio ir fizinio lygmenų;
11. **Elgsena.** Aprašoma sistemos elgsena (galimos jos būsenos ar pan.);
12. **Techniniai ribojimai.** Dėl sistemos skaidymų (arba kitokių ribojimų) gali atsirasti techninių ribojimų, kurie gali daryti įtaką komponentų modeliams;
13. **Funkciniai reikalavimai KFS lygmenyje.** Detaliau aprašomi funkciniai reikalavimai, loginius vienetus išskirstant į kibernetinius ir fizinius komponentus;
14. **Veiklos** (angl. tasks). Užduotys iš pirmojo lygmens yra išskaidomos į veiklas. Joms įgyvendinti yra priskiriami atitinkami komponentai iš fizinio ar kibernetinio pasaulio;
15. **Komponentų sąrašas KFS lygmenyje.** Komponentai yra tipizuojami parenkant jų topologiją. Skiriamos kibernetinės ir fizinės sudedamosios dalys.

Žinant kokias kibernetinių-socialinių sistemų savybes reikia aprašyti, galima parinkti joms modeliuoti pakankamas kalbas, kurios kiekvieną metodo aspektą padengtų bent viena diagrama.

Skirtingai nei kitose reikalavimų analizės ir specifikavimo metodikose, probleminiuose freimuose yra taikomas turinio atskyrimo principas. Kuriamą sistemą yra atskiriama nuo pasaulio, kuriame ji egzistuoja. Tai reiškia, kad yra gaunami du aprašymai. Pirmasis aprašymas pateikia informaciją apie dalykinę sritį ir joje esančias problemas, antrasis – problemai išspręsti reikalingą mašiną (kuriamą sistemą ir jos reikalavimus). Sudaromi artefaktai aprašo sistemos poveikį, o ne pačią sistemą, todėl tai neriboja sistemos įgyvendinimo paradigmos pasirinkimo. Dėl to, dalykinės srities analizei atlikti parenkami probleminiai freimai.



Kibernetiniame ir fiziniame lygmenyje reikia naudoti SysML, kuri leidžia aprašyti sistemą išskaidant ją į atitinkamus pasaulius. Geresniam suderinamumui tarp lygmenų užtikrinti, SysML yra naudojama ir loginių vienetų lygmenyje. 11 pav. pateikiamas reikalavimų pavyzdinis modelis (metodinio karkaso egzempliorius), kurio rezultatų detalumas priklauso nuo jį taikančio sistemos kūrėjo poreikio.



11 pav. KFSS reikalavimų pavyzdinis modelis

11 pav. pateikto KFSS reikalavimų pavyzdinio modelio paaiškinimai:

1. **Probleminiai freimai.** Skirti išanalizuoti ir aprašyti domeną (-us).
2. **Paketų diagrama.** Išskaido probleminiuose freimuose minimas mašinas į smulkesnius loginius paketus;
3. **SysML reikalavimų diagrama loginių vienetų lygmenyje.** Aprašo reikalavimus, keliamus kuriamos sistemos komponentams;
4. **Socialinis modelis.** Kiberfizinės-socialinės sistemos su žmogumi gali bendrauti tiek fiziškai, tiek kibernetiškai (pvz., susirašinėjimo pagalba), bet tarpusavyje KFS bendrauja tik kibernetiniame lygmenyje. Tai reiškia, kad socialinį tinklą galima modeliuoti taip pat, kaip ir agentinėje sistemoje. Užduočių dalinimasis gali būti arba tiesioginis tarp dviejų sistemų, arba per bendrą pranešimų erdvę. Pasirinkus pastarąją, konteksto diagramose turi atsirasti atitinkamas domenai, o šios erdvės savybės ir elgesys turi būti aprašomas probleminiais freimais. Komunikavimui tarp sistemų aprašyti galima naudoti „Foundation for Intelligent Physical Agents“.
5. **SysML blokų diagrama.** Paketų diagrama išskaidoma į detalesnius blokus;

6. **Sekų diagrama.** Modeliuojamas blokų komunikavimas ir sąveika tarp jų;
7. **Būsenų diagrama.** Žinant blokų elgseną galima aprašyti blokų būsenas;
8. **SysML reikalavimų diagrama KFS lygmenyje.** Atskirai aprašo reikalavimus, keliamus kuriamos sistemos kibernetiniams ir fiziniams komponentams. Komponentų pavadinimai paimami iš SysML blokų diagramos;
9. **SysML veiklos diagrama.** Detaliau aprašo užduočių modelį;
10. **SysML vidinės blokų diagramos.** Detalizuojami blokai iš loginio lygmens, atskiriant kibernetinį ir fizinį pasaulius.

### **3.1. Probleminiai freimai**

Nors tradicinio probleminio domeno aprašo užtenka kiberfizinį-socialinių sistemų funkcionalumui aprašyti, darbe yra siūloma visas diagramas papildyti konteksto esybe. Ji leidžia sumažinti kuriamų diagramų kiekį, taip palengvinant ne tik analizės, bet ir projektavimo, procesą.

Remiantis M. Jackson [Jac01], norint pritaikyti probleminius freimus, reikia atlikti šiuos veiksmus:

1. Sukurti konteksto diagramą;
2. Atpažinti ir aprašyti kiekvieną probleminį freimą:
  - a. Identifikuoti vidinius subfreimus;
  - b. Freimą (arba vidinius subfreimus) aprašyti kuo detaliau, naudojantis papildytais probleminių freimų taisyklėmis (3.1.1 – 3.1.5 punktuose);
  - c. Aprašyti subfreimų tarpusavio ryšius.

Toliau yra pateikiami probleminiai freimai ir jų praplėtimo aprašymai.

#### **3.1.1. Sistema-įrankis**

Šiuo probleminiu freimu (žr. 12 pav.) analizuojama problema, kai naudotojams reikalingas įrankis informaciniams objektams kurti ir modifikuoti. Reikia sukurti mašiną, kuri priimdama prašymą vykdyti operaciją, pakeistų reikiamą artefaktą pagal operacijos apibūdinimą. Diagramoje pavaizduotas taškas reiškia, kad artefaktas yra įrankio sudedamoji dalis.

KFSS atveju operacijos rezultatas gali priklausyti nuo konteksto. Kontekstas gali būti interpretuojamas kaip dalykinėje srityje egzistuojantis artefaktas, kuris gali būti sukurtas gamtos arba kažkokios sistemos. Kaip pasiekti ir panaudoti šį artefaktą reikia aprašyti operacijų aprašuose. Analizės metu turi būti aprašyta kuo daugiau galimų priklausomybių ir būdų joms gauti.



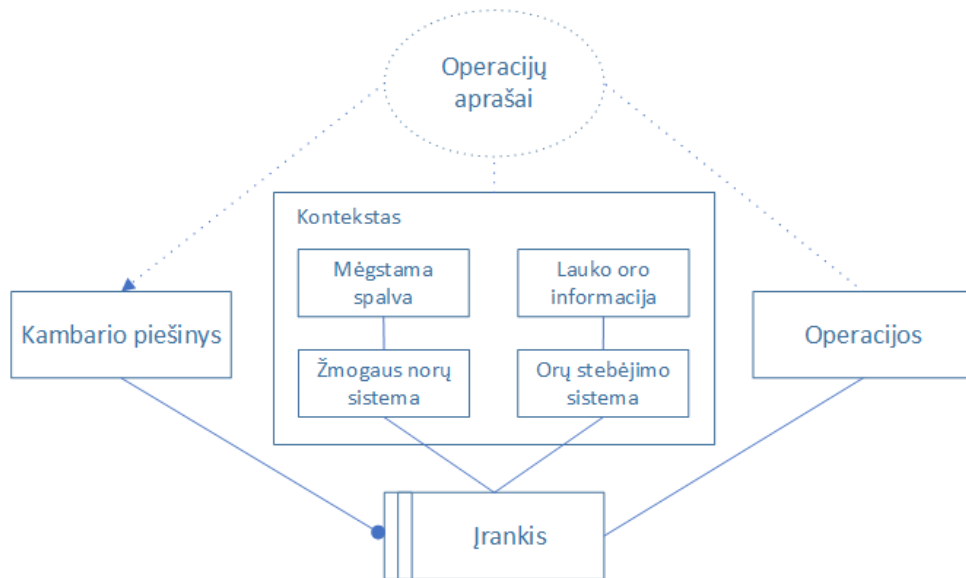
12 pav. Probleminis freimas „Sistema-įrankis“

1 lentelėje yra pateikiamos galimos modeliavimo technikos probleminio freimo esybėms aprašyti.

1 lentelė. Probleminio freimo „Sistema-įrankis“ informacijos modeliavimo technikos

Informacija	Modeliavimo technikos
Įrankio struktūra	Backus-Naur forma, struktūros diagrama,
Operacijų aprašai	Tekstas, baigtinė būsenų diagrama, užduočių diagrama, sprendimų lentelė, tikslų medis, SysML veiklos diagrama
Konteksto informacija ir jo struktūra	Tekstas, SysML blokų aprašų diagrama, SysML vidinė bloko diagrama

Panagrinėkime situaciją, kai koreguojamas artefaktas yra kambario piešinys, o operacija – pridėti spintelę. Tarkime operacijos metu naudotojas nenurodo norimos spintelės spalvos, kas lemia tai, jog įrankis turėtų parinkti žmogui labiausiai patinkančią spalvą. Tokios išvados apie žmogų yra konteksto dalis. Arba jeigu vaizdas už lango turi atitikti dabartinį vaizdą, tai lauko oras būtų dar viena konteksto dalis. Gauti minėto konteksto artefaktus galima iš atitinkamų sistemų (žr. 13 pav.). Diagramoje pavaizduotas taškas reiškia, kad kambario piešinys egzistuoja ir yra valdomas tik įrankyje.



13 pav. Probleminio freimos „Sistema-įrankis“ pavyzdys

### 3.1.2. Valdoma sistema

#### 3.1.2.1. Taisyklėmis valdoma sistema

Šis probleminis freimas analizuoja problemas, kurioms dalykinei sričiai valdyti reikalinga sistema. Reikia sukurti mašiną, kuri sąveikaudama su valdomu elementu užtikrintų reikiamą jo elgseną. Verta paminėti, kad valdomas elementas negali būti statiškas ar pilnai autonominis, nes jis nėra valdomas. Kiberfizinių-socialinių sistemų atveju, toks elementas gali būti dviejų tipų: įprastinis ir sudėtinis. Įprastiniai elementai yra tokie, kurių vidinė struktūra yra nesvarbi, o jiems aprašyti užtenka M. Jackson paminėtų būdų. Tuo tarpu sudėtiniai elementai yra sudaryti tiek iš programų sistemų, tiek ir iš aparatūrinio lygmens. Probleminiame freime šio tipo esybės galima pažymėti žvaigždute (žr. 14 pav.), o juos aprašant naudoti SysML blokų diagramas. Toks detalizavimas leidžia detaliau aprašyti duotų sistemų integravimą, kas palengvina kiberfizinių-socialinių sistemų kūrimą vėlesnėse stadijose.



14 pav. Probleminis freimas „Taisyklėmis valdoma sistema“

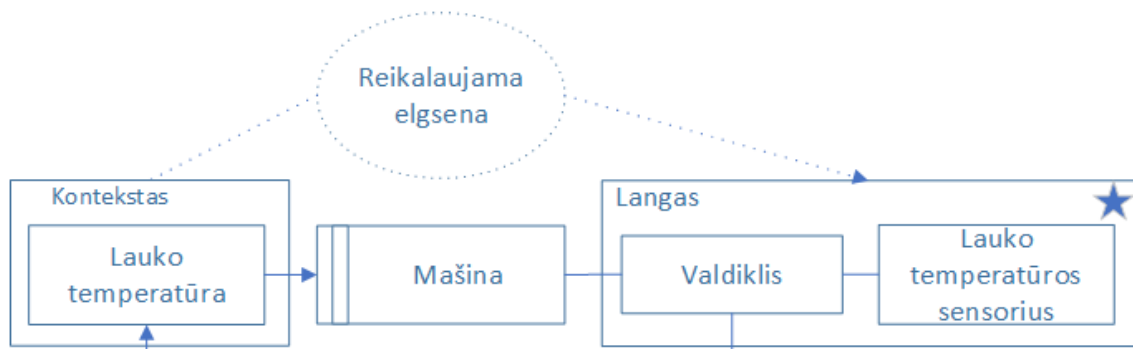
Kadangi reikalaujama elgsena turi būti pritaikyta pagal žmogaus poreikius, tam modeliuoti galima naudoti tikslų medžius. Taip pat kaip ir sistemos-įrankio atveju, jeigu reikalaujama elgsena priklauso nuo konteksto, reikia paminėti kaip jį pasiekti ir panaudoti.

2 lentelėje yra pateikiamos galimos modeliavimo technikos probleminio freimo informacijai aprašyti.

2 lentelė. Probleminio freimo „Taisyklėmis valdoma sistema“ informacijos modeliavimo technikos

<b>Informacija</b>	<b>Modeliavimo technikos</b>
Duomenų modeliai (jeigu yra)	Esybių ryšių diagrama
Valdomo elemento subdomenai, jų ypatybės ir ribojimai. Pavyzdžiui, įstatymai arba jų būsenos, generuojami įvykiai	Tekstas, esybės gyvavimo istorija (angl. Entity Life History), baigtinė būsenų diagrama, sprendimų lentelė, SysML blokų aprašų diagrama, SysML vidinė bloko diagrama
Įvykiai per kuriuos mašina stebės valdomą elementą	Tekstas (galimų įvykių sąrašas)
Veiksmai, kuriuos mašina galės atlikti valdomame elemente	Tekstas (galimų veiksmų sąrašas)
Galimi sutrikimai arba užlaikymai tarp esybių. Pavyzdžiui, valdomas elementas kartais gali būti nepasiekiamas 10 sekundžių	Tekstas
Elgsenos taisyklės	Tekstas, baigtinė būsenų diagrama, sprendimų lentelė, tikslų medis
Konteksto informacija ir jo struktūra	Tekstas, SysML blokų aprašų diagrama, SysML vidinė bloko diagrama

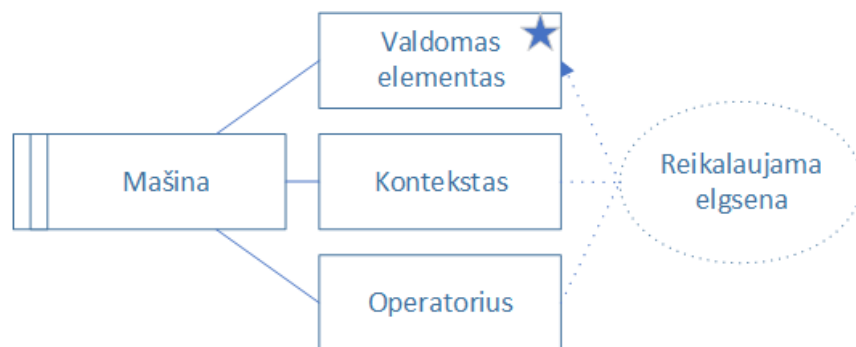
Pavyzdžiui, jeigu pateikiama tokia problema („Jeigu patalpoje esanti temperatūra pakyla aukščiau nei 25 laipsniai, turi atsідaryti langas“), ji gali būti sumodeliuota kaip pavaizduota 15 pav. Valdomas elementas yra langas, o jo sudedamosios dalys yra valdiklis (skirtas komandoms priimti) ir lauko temperatūros sensorius. Kontekstas šiuo atveju yra lauko temperatūra. Reikalaujamoje elgsenoje papildomai turėtų atsirasti tokia informacija: „Jeigu lauko temperatūra yra didesnė nei norima temperatūra, langas neturi būti atidaromas. Lauko temperatūrą galima sužinoti per lango valdiklį“.



15 pav. Probleminio freimo „Taisyklėmis valdoma sistema“ pavyzdys

### 3.1.2.2. Komandomis valdoma sistema

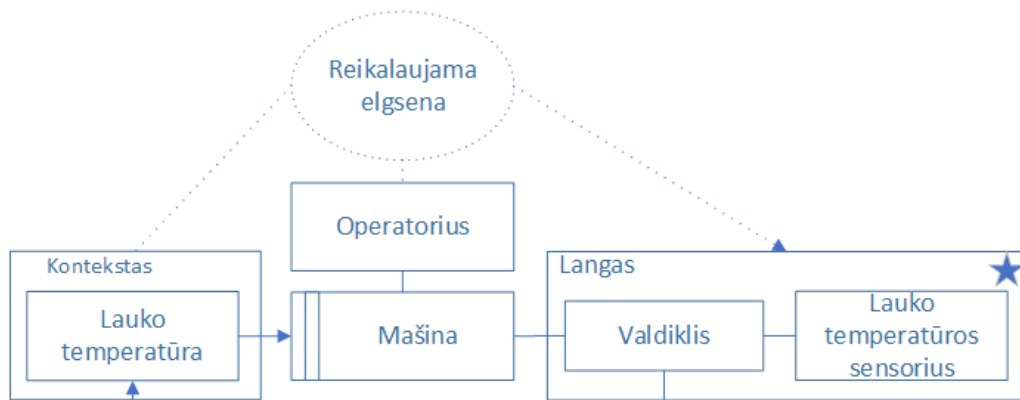
Komandomis valdomos sistemos probleminis freimas modeliuoja problemą (žr. 16 pav.), kai valdomas elementas yra netiesiogiai valdomas operatoriaus. Skirtumas tarp šio freimo ir aukščiau paminėto (taisyklėmis valdomos sistemos) yra tas, kad taisyklėmis valdomoje sistemoje valdomas elementas yra valdomas automatiškai. Tuo tarpu komandomis valdomoje – tik operatoriaus prašymu.



16 pav. Probleminis freimas „Komandomis valdoma sistema“

Komandomis valdomo probleminio freimo esybės gali būti aprašomos pagal 2 lentelėje siūlomas informacijos modeliavimo technikas, papildant elgsenos taisykles komandų, kurios gali būti atliekamos, aprašais.

Pavyzdžiui, esant tokiai situacijai („Vartotojui liepus, langas turi atsidaryti“, žr. 17 pav.), valdomas elementas yra langas, kontekstas – lauko temperatūra, operatorius – naudotojas. Reikalaujama elgsena gali būti papildoma tokia informacija: „Jeigu naudotojui yra labai karšta, o lauko temperatūra yra didesnė nei norima temperatūra, langas neturi būti atidaromas, bet įjungiamas kondicionierius. Lauko temperatūrą galima sužinoti per lango valdiklį“.



17 pav. Probleminio freimo „Komandomis valdoma sistema“ pavyzdys

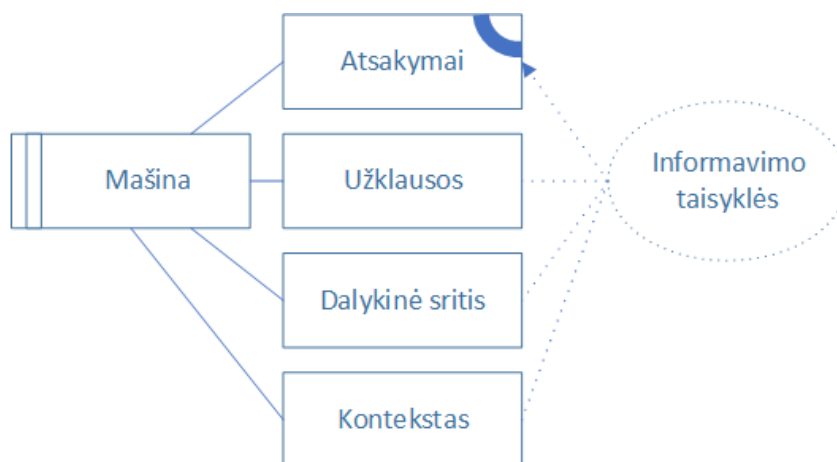
### 3.1.3. Informavimo sistema

#### 3.1.3.1. Diskretaus informavimo sistema

Prereikusi sistema, kuri atsakytų į užklausas apie dalykinę sritį, naudojamas diskretaus informavimo sistemos probleminis freimas. Informacijos šaltinis yra dalykinė sritis, kuri dažniausiai yra suprantama kaip autonominė, bet gali būti ir statinė. Nepaisant to, reikia žinoti jos duomenų struktūrą ir programų sistemoje sukurti atitinkamą duomenų modelį. Informavimo taisyklės apibrėžia atsakymų pavidalą.

Kaip ir anksčiau paminėtų freimų atveju, atsakymai gali priklausyti nuo konteksto, o tokią jų priklausomybę reikėtų aprašyti informavimo taisyklėse.

KFSS atveju atsakymai gali būti įprastiniai ir specifiniai. Įprastiniai atsakymai yra tokie, kuriuos sugeneravus jie yra laikomi mašinoje arba perduodami nurodytoms sistemoms. Tuo tarpu specifinio tipo atsakymai turi ne tik įprastinio savybes, bet taip pat yra panaudojami mašinos apmokymui. Kadangi tai yra viena iš kiberfizinųjų-socialinių sistemų savybių, todėl šio tipo atsakymai turi būti pažymėti papildoma juostele kampe (žr. 18 pav.). Taip modeliuojamas probleminis freimas atkreipia sistemos kūrėjų dėmesį vėlesnėse gyvavimo ciklo stadijose.



18 pav. Probleminis freimas „Diskretaus informavimo sistema“

Taip pat dėl kiberfizinio-socialinių sistemų specifikos, atsakymų esybę gali sudaryti vidiniai komponentai, kurie gali būti tiek kibernetiniai, tiek fiziniai. Pavyzdžiui, tai gali būti koks nors HTTP būdu pasiekiamas servisas, ekranas arba garso kolonėlės (žr. 19 pav.).

3 lentelėje yra pateikiami probleminio freimo informacijos modeliavimo technikų pasiūlymai.

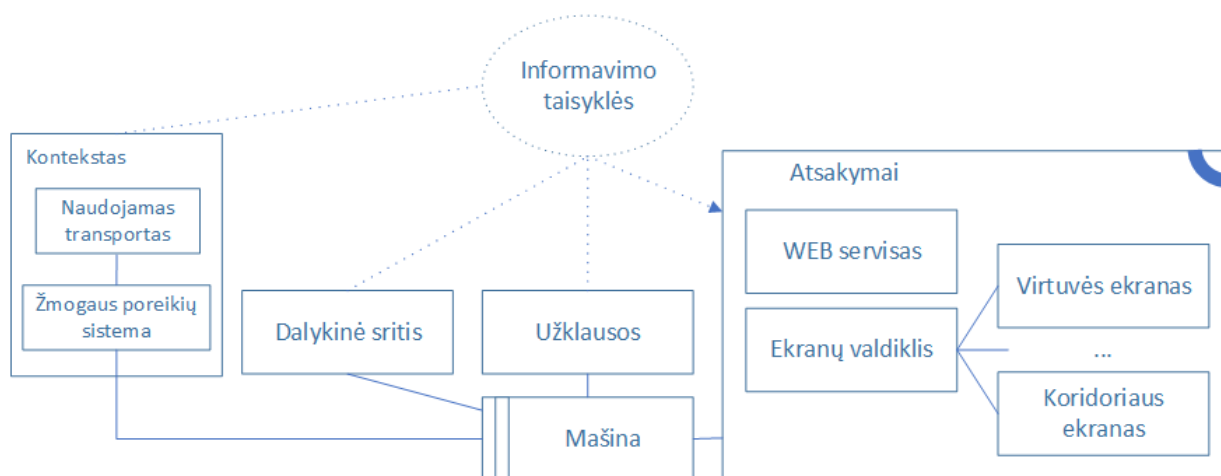
3 lentelė. Probleminio freimo „Diskretaus informavimo sistema“ informacijos modeliavimo technikos

Informacija	Modeliavimo technikos
Dalykinės srities duomenų modelis	Esybių ryšių diagrama
Kiekvieno dalykinės srities subdomeno charakteristika, įskaitant įvykius, kurie gali pakeisti probleminės srities būseną. Jei įmanoma, reikėtų nurodyti galimus įvykių eiliškumus	Tekstas (galimų įvykių sąrašas), esybės gyvavimo istorija
Kaip mašina gali pasiekti informaciją apie probleminės srities būseną ir ten vykstančius įvykius	Tekstas
Galimi sutrikimai arba užlaikymai tarp esybių	Tekstas
Informacija apie tai, iš kur gauti duomenis reikalingus darbo pradžiai (inicijavimui)	Backus-Naur forma, struktūros diagrama, failų išsidėstymas
Informavimo taisyklės. Pavyzdžiui, kaip turi atrodyti atsakymai ar kokios užklausa turi būti palaikomos	Tekstas, įvairios diagramos, lentelės, tikslų medis
Konteksto informacija ir jo struktūra	Tekstas, SysML blokų aprašų diagrama, SysML vidinė bloko diagrama



Mašinos apmokymo taisyklės	Tekstas, SysML veiklos diagrama, sprendimų lentelė
----------------------------	--

Pavyzdžiui, panagrinėkime pateiktą problemą: „Naudotojas gali paklausti apie orą lauke“. Taigi, užklauskos gali būti įvairios: apie lauko temperatūrą, ar lyja, ar sninga, ar saulėta ir panašiai. Atsakymai, priklausomai nuo naudotojo vietos, gali būti pateikiami per garso kolonėles arba kelis, esančius arčiausiai jo, ekranus. Papildoma juostelė prie atsakymų reiškia, kad mašina turi panaudoti šį atsakymą savęs apmokymui. Turint užtektinai apmokymo duomenų, mašina gali pateikti atsakymus greičiau, o ryšio sutrikimai tarp mašinos ir dalykinės srities gali būti neapastebimi. Konkrečios situacijos nuspėjimo pavyzdys galėtų būti toks, kad jeigu 10 dienų iš eilės sninga ir dabar yra žiemos vidurys, tikėtina, kad snigs ir šiandien. Tai reiškia, kad kuriama mašina turi mokėti ne tik atsakyti į užduodamus klausimus, bet ir nuspėti jų atsakymus. Šiuo atveju konteksto pavyzdys galėtų būti dažniausiai naudojama naudotojo transporto priemonė. Jeigu tai yra viešasis transportas, atsakymas galėtų būti toks: „Šiandien snigs visą dieną, dėl to galimi autobusų vėlavimai. Šiandien turite susitikimą 12:00, dėl to vertėtų išvykti anksčiau, pavyzdžiui, 52 autobusu, kuris stotelėje bus 10:25“.



19 pav. Probleminio freimo „Diskretaus informavimo sistema“ pavyzdys

### 3.1.3.2. Nuolatinio informavimo sistema

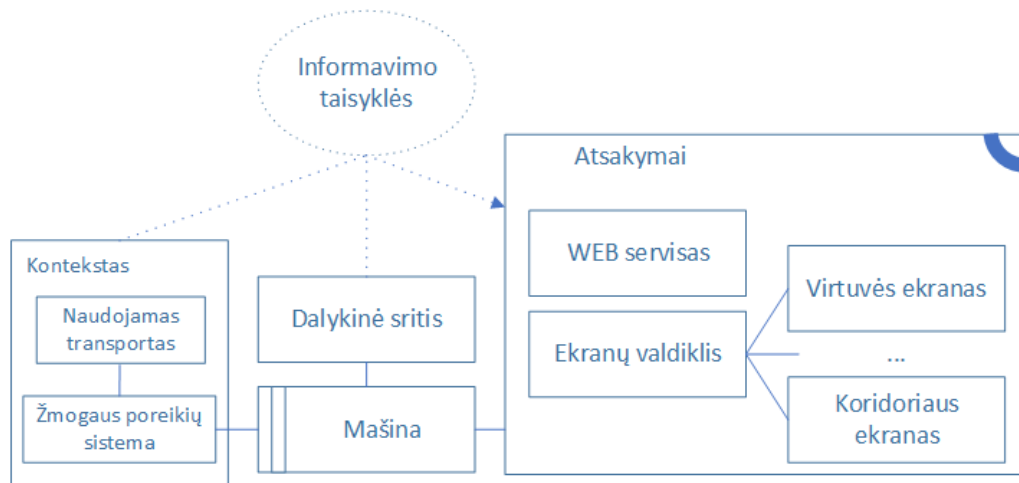
Nuolatinio informavimo sistemos probleminis freimas (žr. 20 pav.) modeliuoja situaciją, kai reikalinga sistema, kuri nuolat pateikia informaciją apie dalykinę sritį. Kaip ir diskretaus informavimo probleminio freimo atveju, esybė „teikiama informacija“ gali būti dviejų tipų ir kampe žymima juoste.



20 pav. Probleminis freimas „Nuolatinio informavimo sistema“

Nuolatinio informavimo probleminio freimo esybės gali būti aprašomos pagal 3 lentelėje siūlomas informacijos modeliavimo technikas, tačiau tai atliekant būtina atsižvelgti į tai, kad šio probleminio freimo atveju informavimo taisyklių modeliavimui nereikalingas palaikomų užklausų aprašymas.

Žemiau pateikiamas nuolatinio informavimo probleminio freimo pavyzdys (žr. 21 pav.).



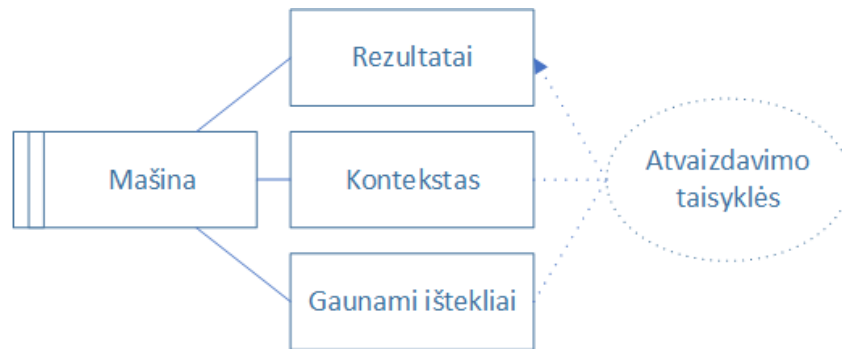
21 pav. Probleminio freimo „Nuolatinio informavimo sistema“ pavyzdys

### 3.1.4. Transformavimo sistema

Jei reikalinga sistema, kuri gaudama išteklius sugeba juos apdoroti, tam naudojamas transformavimo sistemos probleminis freimas. Šis freimas taikomas problemoms, kurios būna dviejų tipų:

1. Žinomi pradiniai duomenys, kurie turi būti pakeisti norint gauti rezultatą;
2. Gaunami duomenys yra informacinės esybės, o mašina turi juos transformuoti.

Gaunami ištekliai turi būti statiškas dalykinės srities elementas, rezultatus keičia tik mašina, o atvaizdavimo taisyklės nusako įvesties-išvesties sąryšį. Kaip ir informavimo sistemose, KFSS atveju turėtų atsirasti aprašas apie galimas priklausomybes nuo konteksto (žr. 22 pav.).



22 pav. Probleminis freimas „Transformavimo sistema“

Siūlomos informacijos aprašymo technikos pateikiamos 4 lentelėje.

4 lentelė. Probleminio freimo „Transformavimo sistema“ informacijos modeliavimo technikos

Informacija	Modeliavimo technikos
Gaunamų išteklių ir rezultatų duomenų aprašai	Backus-Naur forma, struktūros diagrama
Duomenų gavimo šaltiniai ir rezultatų pateikimo informacija	Tekstas
Atvaizdavimo taisyklės	Tekstas, sąryšių lentelė, struktūros diagrama, tikslų medis
Konteksto informacija ir jo struktūra	Tekstas, SysML blokų aprašų diagrama, SysML vidinė bloko diagrama

Pirmojo problemos tipo atveju, pavyzdys galėtų būti dokumento skenavimo mašina. Gaunamas išteklius – popieriaus lapas, rezultatas – popieriaus lapo paveikslukas. Mašinos užduotis – iš realaus lapo jį transformuoti į virtualų. Priklausomybė nuo konteksto galėtų būti skenuojamo teksto patikrinimas, pavyzdžiui, jeigu tekste teigiama, kad šeimininkui patinka raudona spalva, bet žinoma, kad tikroji patinkanti spalva yra mėlyna, mašina gali pabraukti šį sakinį ir komentare parašyti pataisymą.

Antros problemos tipo pavyzdžiu, galėtų būti mašina, kuri apskaičiuoja atlyginimus. Gaunamas išteklius būtų darbuotojų užpildytos laiko lentelės ir užduočių valandinis mokestis (skaitmeniniame formate), o rezultatas – apskaičiuoti atlyginimai. Mašinos užduotis – iš laiko lentelių, pagal užduočių valandinį mokesį, apskaičiuoti darbuotojo atlyginimą. Priklausomybė nuo konteksto – statybų darbo metu buvusios oro sąlygos. Kuo jos buvo sudėtingesnės, tuo labiau turi didėti valandinis įkainis.

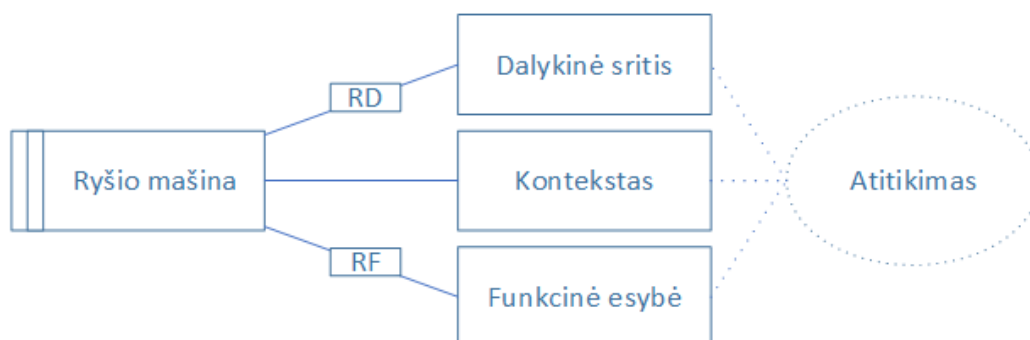
### 3.1.5. Ryšio sistema

Atlikti probleminės srities analizę būtų žymiai paprasčiau, jeigu ryšys tarp esybių visada būtų tiesioginis. Tačiau neretai pasitaiko, kad bendriems ryšiams (ar jų patikimumui) užtikrinti, reikia sukurti tarpininką. Tarpininko sukūrimo problema yra analizuojama naudojantis ryšio sistemos probleminiu freimu.

Kaip ir kituose probleminiuose freimuose, kiberfizinės-socialinės sistemos atveju reikia aprašyti priklausomybes nuo konteksto.

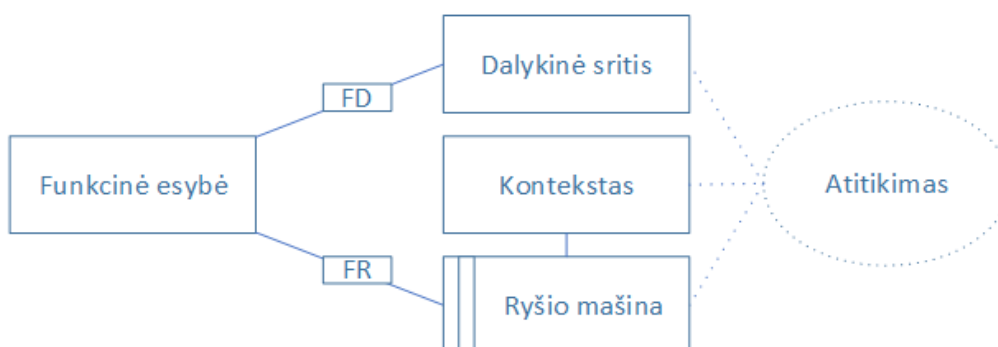
Gali būti išskiriami du tarpininkų tipai:

1. Tarpininkas, užtikrinantis ryšį tarp dalykinės srities ir funkcinės esybės (žr. 23 pav.);
2. Tarpininkas, kuris inicijuoja ir palaiko ryšį tarp funkcinės esybės ir sistemos (žr. 24 pav.).



23 pav. Ryšio tarp dalykinės srities ir funkcinės esybės užtikrinimo probleminis freimas

Pirmojo tarpininko pavyzdys galėtų būti prisijungimo patvirtinimas naudojantis mobiliuoju telefonu. Dalykinė sritis būtų sistema, prie kurios norima prisijungti, funkcinė esybė – mobilusis telefonas, o ryšio mašina būtų sistema, kurią norima sukurti. Vienas iš ryšio mašinos galimų variantų yra SMS siuntimo sistema. Bendri elementai tarp ryšio mašinos ir dalykinės srities (RD) būtų pranešimas, kad naudotojas nori prisijungti ir kodas, kurį jis turi įvesti. Tuo tarpu bendri elementai tarp ryšio mašinos ir funkcinės esybės (RF) būtų SMS žinutė su kodu. Galimas konteksto variantas – naudotojo prašymas naktimis siųsti žinutę kitu numeriu.



24 pav. Ryšio tarp sistemos ir funkcinės esybės užtikrinimo probleminis freimas

Antruoju atveju, pavyzdys – pacientų stebėjimo sistema, kuri surenka jų būklę stebinčių daviklių duomenis ir nėra pavaizduota probleminiame freime. Dalykinė sritis šiuo atveju yra pacientas, funkcinė esybė – daviklis, o ryšio mašina yra įranga, stebinti ir siunčianti informaciją. Bendri elementai tarp funkcinės esybės ir dalykinės srities (FD) – kūno (daviklio) temperatūra, o tarp funkcinės esybės ir ryšio mašinos (FR) – įtampa. Ryšio mašina gaudama įtampos duomenis gali nuspręsti, kokia yra paciento temperatūra, o jei ji viršija normą (tarkime 45°C), gali sistemai pranešti apie daviklio gedimą. Konteksto pavyzdys galėtų būti paciento veikla, nes miego metu normali kūno temperatūra yra žemesnė.

Galimos informacijos modeliavimo technikos yra nurodomos 5 lentelėje.

5 lentelė. Probleminio freimo „Ryšio sistema“ informacijos modeliavimo technikos

<b>Informacija</b>	<b>Modeliavimo technikos</b>
Galimos būsenos ir įvykiai dalykinėje srityje	Tekstas (galimų įvykių sąrašas), baigtinė būsenų diagrama
Aprašyti duomenų atsparumą (angl. redundancy) dalykinėje srityje. Svarbu paminėti kaip nustatyti, kurie gauti duomenys yra teisingi	Tekstas, sprendimų lentelė
Gaunamos informacijos sąryšių informacija (RD, RF, FR). Pavyzdžiui, gaunama 1V įtampa atitinka 37°C	Lygiagreti (angl. concurrent) baigtinė būsenų diagrama, tekstas, sąryšių lentelė, sprendimų lentelė
Norimos informacijos struktūra	Baigtinė būsenų diagrama, tekstas, sąryšių lentelė, sprendimų lentelė
Konteksto informacija ir jo struktūra	Tekstas, SysML blokų aprašų diagrama, SysML vidinė bloko diagrama

### 3.2. Analizės ir modeliavimo pavyzdinis procesas

1. Aprašoma sistemos vizija;
2. Aprašomi suinteresuoti žmonės;
3. Aprašomi tikslai ir ribojimai;
4. Sukuriama konteksto diagrama;
5. Sukuriamas tikslų modelis;
6. Sukuriamas užduočių modelis;
7. Iš konteksto diagramos ir užduočių modelio sukuriamas probleminis freimas (arba keli sub-freimai);
8. Aprašomi probleminiai freimai;
9. Dėl kylančių problemų freimai papildomi naujais aprašais. Galimi šių problemų pavyzdžiai ir sprendimai:
  - a. Dvi KFS įkėlė po vieną tikslą į bendrą erdvę. Šie tikslai prieštarauja vienas kitam. Reikia aprašyti, koku būdu bus nuspręsta, kurios sistemos tikslas bus atmestas. Pavyzdžiui, viena KFS nori atidaryti kvartalo vartus, kita – uždaryti. Kuris variantas bus pasirinktas? Greičiausiai, tokius prieštaravimus sprendžia bendro pranešimo erdvė arba kiekviena KFS atskirai. Bet koku atveju, sprendimas tokiose situacijose bus teikti pirmenybę atidarymui,

o kitą tikslą arba ignoruoti, arba įvykdyti po atidėto laiko. Pastaruoju atveju reikės aprašyti kaip apskaičiuojamas atidėjimas bei leisti tai sistemai nuspręsti ar ji nori to atidėjimo;

- b. Viena KFS sistema pateikė užduotį, kitos sistemos jį priėmė ir pradėjo vykdyti, bet per nustatytą laikotarpį rezultato negavo (arba jis vis dar netenkinamas). Reikia aprašyti, koks procesas vyksta su tokia užduotimi. Pavyzdžiui, viena sistema paprašė pagalbos skaičiavimams vykdyti. Kitos sistemos prašymą priėmė ir įvykdė, išskyrus vieną sistemą, kuri per nustatytą laikotarpį rezultato nepateikė. Reiktų nuspręsti ar daliniai rezultatai vis dar naudingi, ar reikia kartoti visos užduoties vykdymą iš naujo. Taip pat ar sistema gali atsiųsti rezultatą kai ji atsiras.

10. Detalizuojama pirmoji konteksto diagrama;
11. Sukuriama paketų diagrama pagal probleminį freimą (ar sub-freimus);
12. Aprašomi reikalavimai naudojant SysML reikalavimų diagramą;
13. Aprašomos socialinio tinklo taisyklės;
14. Sukuriamos arba papildomos SysML blokų diagramos;
15. Sukuriamos arba papildomos sekų diagramos;
16. Sukuriamos arba papildomos būsenų diagramos;
17. Konteksto diagrama detalizuojama išskaidant jos komponentus į kibernetinį ir fizinių lygmenis;
18. Aprašomi (ir jeigu reikia vėliau atnaujinami) techniniai ribojimai;
19. Detalizuojama reikalavimų diagrama, išskiriant jos komponentus į kibernetinį ir fizinių lygmenis;
20. Sukuriamos arba papildomos SysML veiklų diagramos. Dėl socialinio modelio, taip pat gali atsirasti ir naujos veiklos (pavyzdžiui, padėti kitiems nariams atlikti skaičiavimus ar pateikti kokią nors informaciją, nes tik vienas narys turi reikiamą jutiklį ir panašiai);
21. Sukuriamos SysML vidinių blokų diagramos.

Apibendrinant, darbe yra pasiūlytas reikalavimų metodinis karkasas, tinkamos specifikavimo kalbos, taip pat sukurtas pavyzdinis reikalavimų modelis. Jame probleminių freimų pagalba, skirtingai nei kitose analizės ir specifikavimo metodikose, yra pateikiamas probleminio domeno aprašas ir sistemos poveikis išorei. Žemesniuose lygmenyse SysML pagalba sistema yra detalizuojama, galiausiai išskaidant jos vidinę struktūrą į kibernetinius ir fizinius pasaulius. Sistemų tarpusavio komunikacijai aprašyti naudojama agentams skirta kalba FIPA-ACL. Gautas rezultatas apibūdina visus svarbiausius sistemos aspektus, paminėtus antrame skyriuje:

1. Sprendžiamas problemas;
2. Reikalavimus;
3. Dalykinę sritį;
4. Sistemos bendravimą su dalykine sritimi, apimant ir naudotojus.

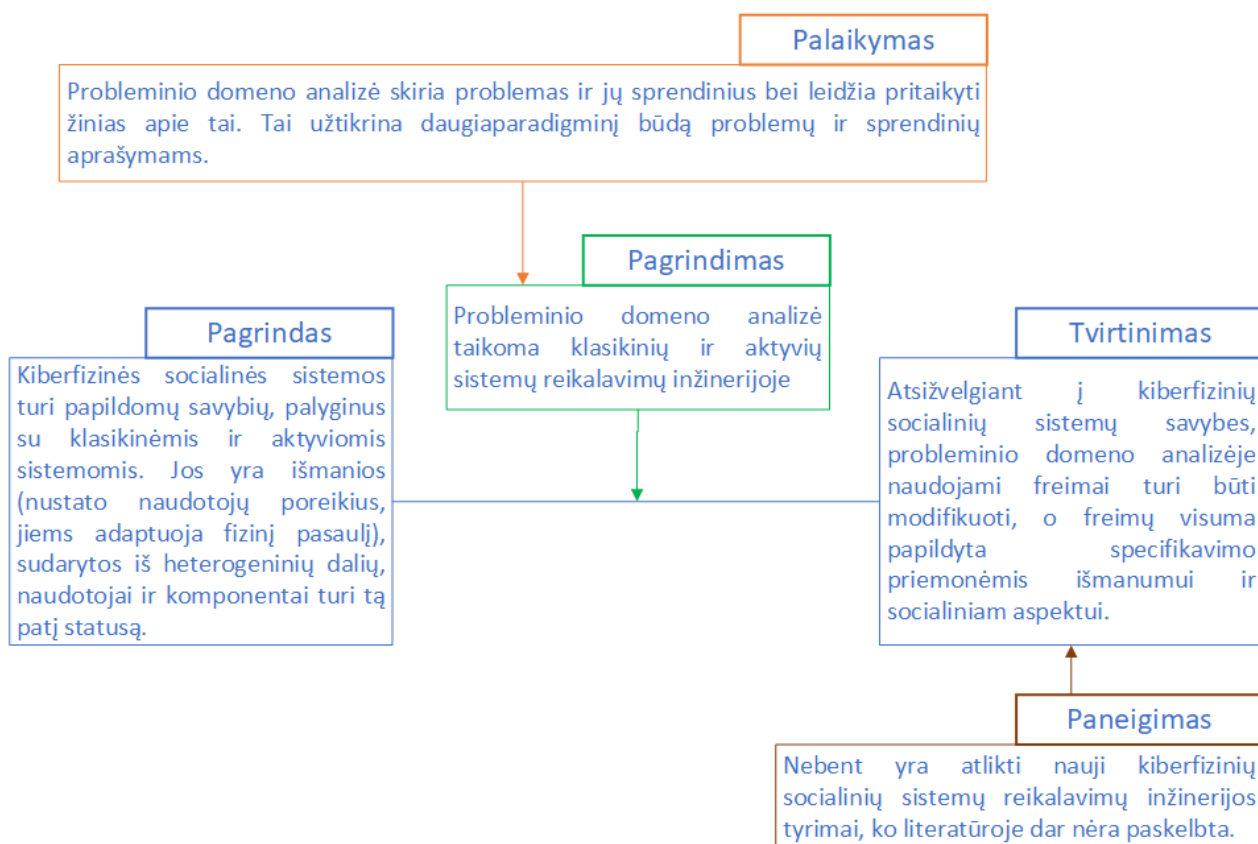
## 4. Metodikos realizacija

Magistrinio darbo ketvirtame skyriuje yra pritaikoma siūloma metodika, remiantis 1.3 poskyriuje aprašytu kiberfizinė-socialinių sistemų poklasiu – išmanių namų bendrija. Atliekant metodikos realizaciją dėmesys skiriamas paskutinio magistrinio darbo uždavinio įvykdymui – pritaikytos probleminės domeno analizės panaudojimui.

Dėl to, kad šios metodikos pritaikymo dalis sutampa su tradicinių sistemų kūrimu, darbe bus nagrinėjami tik žemiau išvardinti, su sistemos lygmens dekomponavimu susiję, artefaktai:

- Konteksto diagrama sistemos lygmenyje, kuri yra vienas iš pritaikytos reikalavimų analizės ir modeliavimo metodikos reikalavimų;
- Konteksto diagrama loginių vienetų lygmenyje, kuri reikalinga tam, kad parodyti, kaip įgyvendinti komponentų atsekamumą žemesniuose lygmenyse.

Rezultatų teisingumas išsamiai gali būti įvertintas vėlesnėse sistemos gyvavimo ciklo stadijose, kurios nebuvo įgyvendintos dėl darbo apimties ribojimų. Kadangi baigiamasis darbas yra argumentacinio pobūdžio, tai rezultatų teisingumas gali būti pagrindžiamas analizuojant argumentavimą. 25 pav. pateikiamas darbe panaudotas argumentas sudarytas pagal Toulmin metodą.

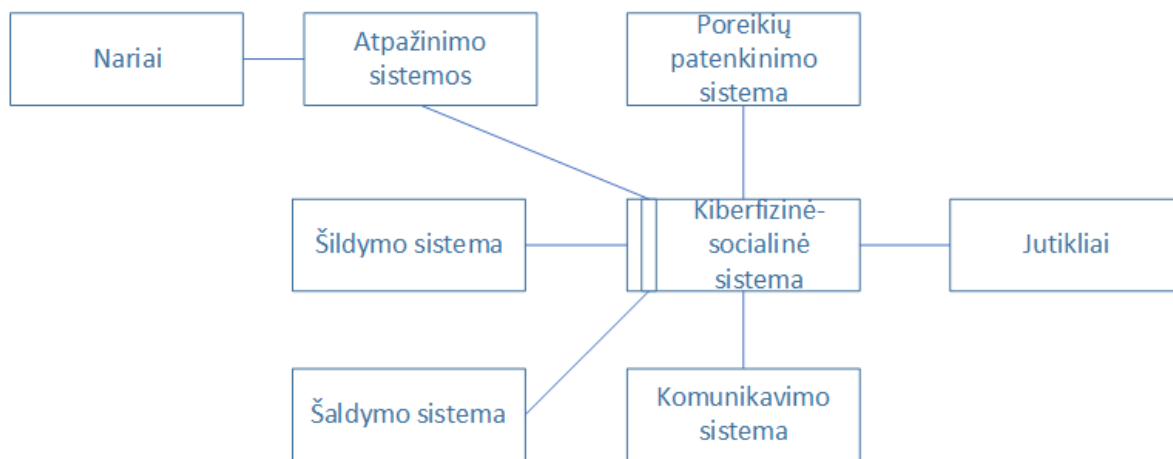


25 pav. Rezultatų teisingumo pagrindimas panaudojant Toulmin metodą

Žemiau pateikiamas atvejis, kuriam bus pritaikytas KFSS reikalavimų pavyzdinis modelis: „Bendrijoje yra tik keli išmanūs namai. Statau kelis naujus namus. Nauji ir esami namai turi

sugebėti tarpusavyje bendradarbiauti. Esminis skirtumas tarp jų yra tas, kad senieji neturi jutiklių lauke. Taip pat reikalingas sprendimas, kuris leistų žmonėms su sistema bendrauti lyg su antru žmogumi, nepririšant jų prie vieno konkretaus įrenginio ar buvimo vietos. Sistema turi atpažinti name gyvenančius, dažnai besilankančius ir nepažįstamus žmones. Namų klimatas turi būti pritaikytas pagal tame būste gyvenančių žmonių norus. Analizuojant žmonių elgesį, klimatą reikia koreguoti pagal jų pageidavimus. Pavyzdžiui, jeigu gyvenantis žmogus kelis kartus apie 21 valandą žemino temperatūrą, tuomet sistema turi pradėti daryti tai pati. Jeigu gyvenantis naudotojas kelis kartus apie 8 valandą padidindavo temperatūrą, tą turi padaryti ir sistema. Dažnai besilankančių žmonių klimatas neturi būti stebimas, tačiau jeigu jiems atvykus visada būdavo paruošiama kava, sistema turi pasidomėti ar ji turi tai atlikti. Šiuo atveju sistema turi taikyti tam tikras išimtis, pavyzdžiui, nakties metu gėrimo nesiūlyti. Nepažįstamų naudotojų elgesys turi būti itin stebimas. Jeigu nepažįstamas naudotojas paėmė ne jam priklausantį daiktą, apie tai turi būti informuoti gyvenantys naudotojai. Tuo tarpu, jeigu jų nėra namuose, o namai palikti atrakinti ir nutinka minėta situacija – papildomai turi užrakinti visas duris, uždaryti langus ir informuoti gretimus namus apie galima vagystę“.

Šiuo konkrečiu atveju konteksto diagrama yra sudaryta iš toliau išvardintų dalių (žr. 26 pav.): poreikių patenkinimo sistemos, komunikavimo sistemos, jutiklių, KFSS, šildymo sistemos, šaldymo sistemos ir atpažinimo sistemos (narių).



26 pav. Sistemos lygmens konteksto diagrama

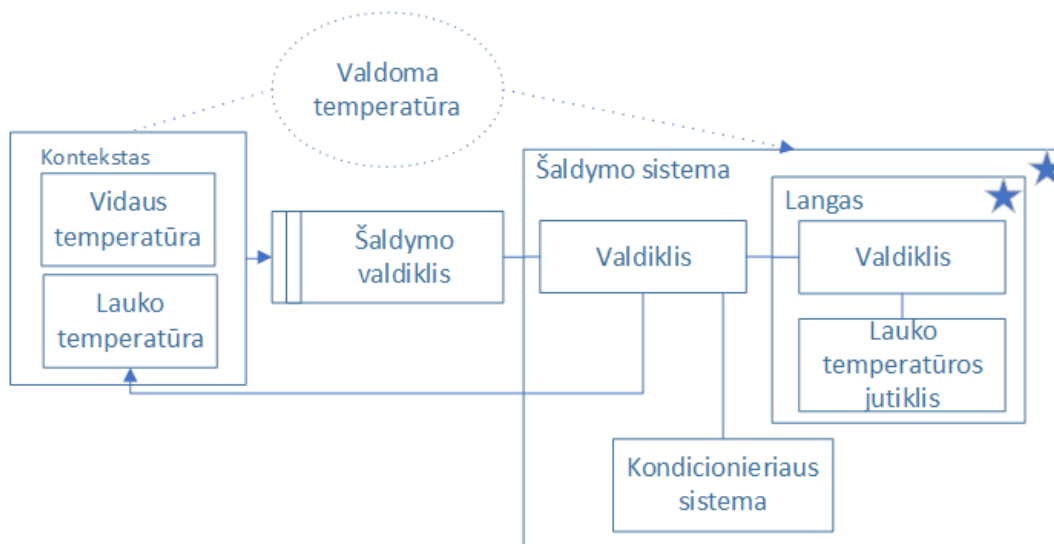
Sistemos lygmens konteksto diagramoje pavaizduota kibertizinė-socialinė sistema yra dekomponuojama probleminių freimų pagalba. Remiantis šia schema yra išskiriami tokie freimai: taisyklėmis valdomos sistemos (šildymo, šaldymo ir poreikių patenkinimo) ir nuolatinio informavimo sistema (komunikavimo).

Šildymo ir šaldymo sistemų probleminiai freimai yra itin panašūs, dėl to darbe pateiktas probleminis freimas tik vienai iš jų – šaldymo sistemai (žr. 27 pav.). Priklausomai nuo situacijos, šaldymo sistema gali jau egzistuoti arba būti kuriama. Kadangi jos sukūrimui užtenka tradicinių



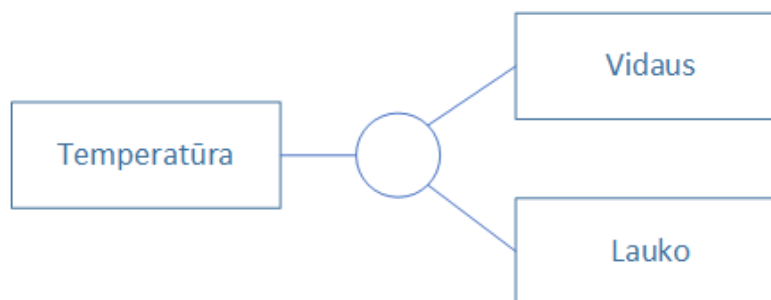
metodikų, darbe bus nagrinėjama situacija, kai šaldymo sistema yra egzistuojanti. Diagramoje ši sistema yra dekomponuojama į smulkesnius komponentus pagal jos aprašą.

Šaldymo sistemos probleminio freimo kontekstas turi pateikti informaciją apie lauko ir vidaus temperatūras. Lauko temperatūros duomenys yra gaunami iš šaldymo sistemos, arba tiksliau, jos posistemės jutiklio pagalba. Tuo tarpu informacija apie vidaus temperatūrą yra gaunama iš šildymo sistemos.



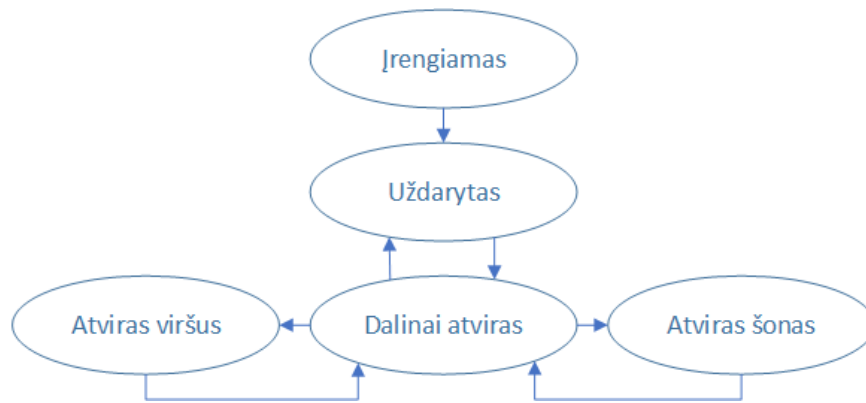
27 pav. Probleminis freimas: taisyklėmis valdoma šaldymo sistema

**Duomenų modelis.** Pateiktos šaldymo sistemos duomenų modelio esybių ryšių diagrama (žr. 28 pav.) yra nesudėtinga, nes sudaryta tik iš vieno elemento – vidinės arba lauko temperatūros.



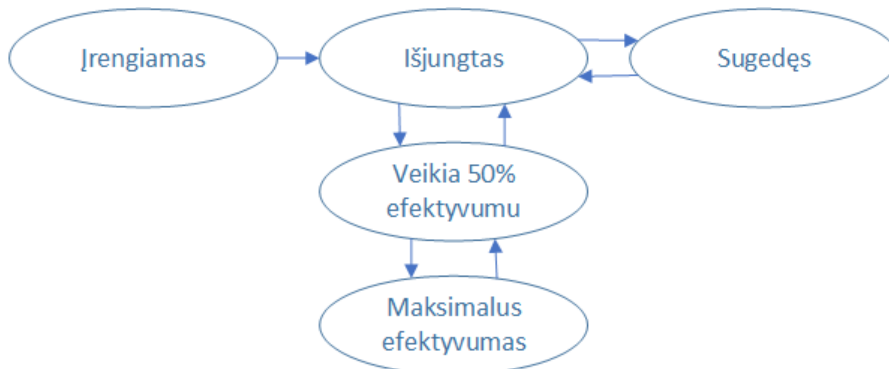
28 pav. Šaldymo sistemos esybių ryšių diagrama

**Valdomas elementas.** Name įrengta šaldymo sistema turi savo atskirą valdiklį, kurio pagalba valdomi langai arba kondicionieriai. Langas gali turėti vieną iš penkių būsenų (žr. 29 pav.). Kai langas yra pilnai įrengtas, jo būseną pasikeičia į uždarytą. Norint langą atidaryti, jis pirmiausia turi pereiti į dalinai atvirą būseną, o tada pasirinktinai arba atverti viršų, arba šoną. Verta paminėti, kad langas neturi užspaudimo apsaugos.



29 pav. Lango būsenos

Kondicionieriui gali būti priskiriama viena iš 5 būsenų (žr. 30 pav.). Kai kondicionierius yra pilnai įrengtas, jo būsena pasikeičia į išjungtą. Tuomet jis gali pereiti į būseną „Sugedęs“ tik tuo atveju, jeigu bandant pakeisti būseną į „Veikia 50% efektyvumu“, ji nepasikeičia per 10 sekundžių. Perėjimą į šią būseną valdo kondicionieriaus vidinis valdiklis.



30 pav. Kondicionieriaus būsenos

**Veiksmi, kuriuos šaldymo valdiklis galės atlikti.** Šaldymo sistemos valdymas vyksta vienintelio valdiklio name pagalba. Galimos žinutės pateiktos 6 lentelėje:

6 lentelė. Valdiklio palaikomų pranešimų formatas

Žinutės formatas	Galimos reikšmės	Rezultatas
state:<identifikacija>:<būsena>	Identifikacija: langas-X, kur X – norimo lango numeris. Kondicionierius-Y, kur Y – norimo kondicionieriaus numeris. Būsena: žiūrėti galimas būsenas pagal 29 pav. ir 30 pav.	Įrenginys pagal <identifikacija> yra nustatomas į būseną <būsena>.

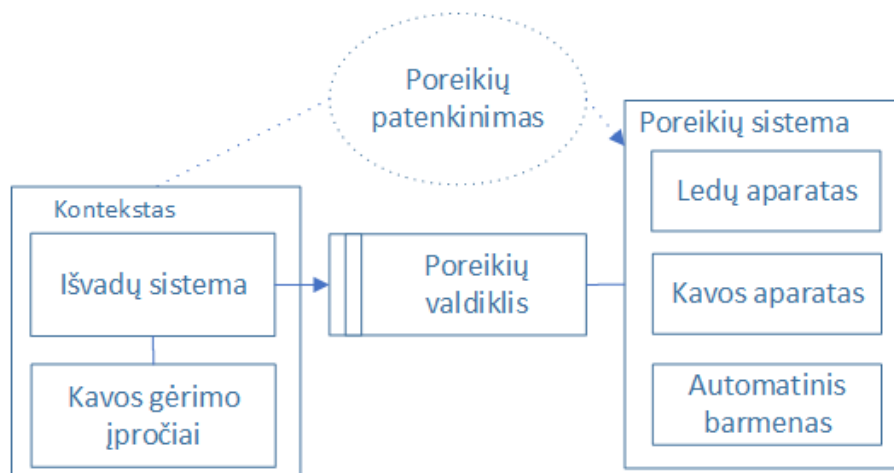
info		Gaunamos visų įrenginių būsenos.
off		Visi įrenginiai yra išjungiami

**Galimi sutrikimai arba užlaikymai.** Kondicionierius pakeičia savo būseną iš „Veikia 50% efektyvumu“ į „Išjungtas“ arba „Maksimalus efektyvumas“ per 10 sekundžių, kurių metu nauji prašymai apie būsenos keitimą yra ignoruojami. Tokį perėjimą užtikrina kondicionieriaus vidinė sistema.

**Elgsenos taisyklės.** Šaldymo valdiklis kas 10 sekundžių turėtų iš konteksto gauti informaciją apie įprastinę tam momentui būdingą temperatūrą ir palyginti ją su vidine. Jeigu esama vidinė temperatūra yra aukštesnė, tuomet valdiklis turi priimti sprendimą, kuris šaldymo būdas bus naudojamas. Jeigu lauko temperatūra yra lygi ar žemesnė už įprastinę vidinę – tuomet langas atidaromas. Kitu atveju – atidaryti langai uždaromi ir įjungiamas kondicionierius.

**Konteksto informacija.** Šiuo atveju kontekstas iš išvadų sistemos pateikia duomenis apie įprastinę vidinę temperatūrą būdingą jos matavimo laikui; iš šaldymo sistemos – apie vidutinę temperatūrą aplink namą (langų jutiklių pagalba); iš šildymo sistemos yra pateikiama kiekvieno kambario vidinė temperatūra.

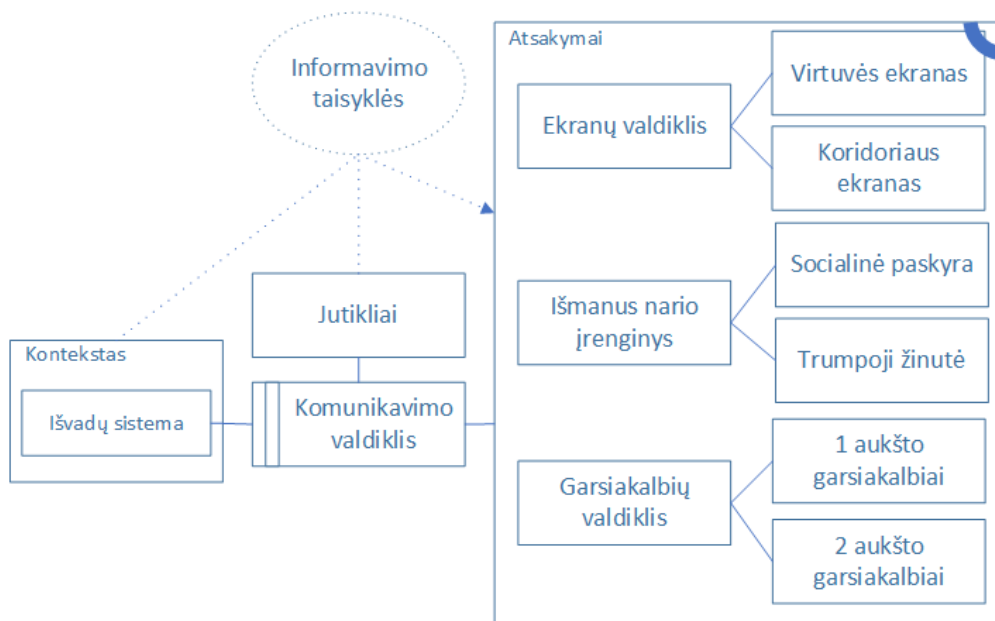
Toliau darbe pateikiami poreikių patenkinimo, komunikavimo ir išvadų sistemų probleminiai freimai, neskiriant ypatingo dėmesio jų detaliai aprašymui. Poreikių patenkinimo probleminio freimo atveju (žr. 31 pav.), poreikių valdiklis turėdamas patenkinti narių poreikius, turi gauti šią informaciją iš kontekste esančios išvadų sistemos. Pastarojoje sistemoje yra aprašomos visos taisyklės pagal kurias turi būti patenkinami poreikiai ir nurodyti būdai šiems veiksams atlikti (pavyzdžiui, dažnai besilankantis narys „X“ mėgsta kavą su pienu, bet be cukraus).



31 pav. Problemnis freimas: taisyklėmis valdoma poreikių patenkinimo sistema

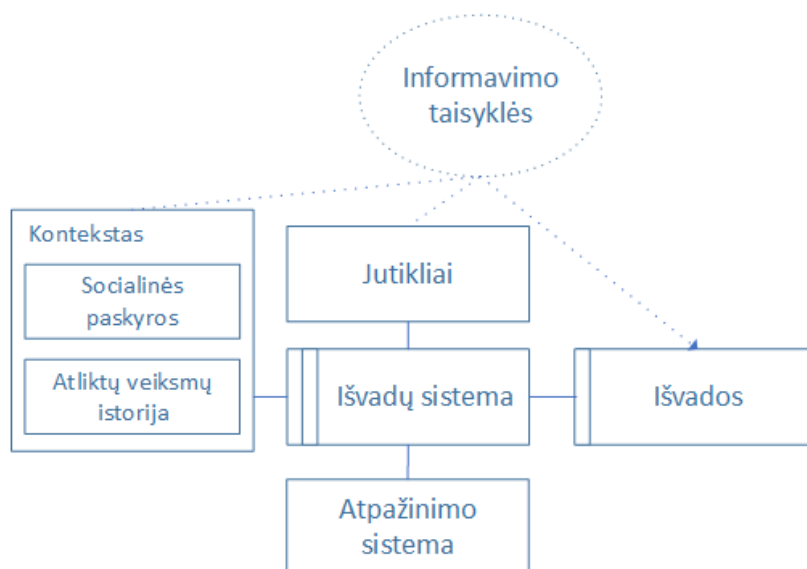
Atsakymų pateikimą nariams pagal kontekste esamą informaciją užtikrina komunikavimo sistema (žr. 32 pav.). Informacija apie dalykinę sritį yra gaunama jutiklių pagalba, o atsakymui

tinkamiausia sistema yra parenkama pagal nario buvimo vietą. Šių sistemų yra trys: virtuvėje ir koridoriuje įrengti ekranų valdikliai, nario telefonas ir dviejuose aukštuose įrengti garsiakalbiai.



32 pav. Probleminis freimas: nuolatinio informavimo komunikavimo sistema

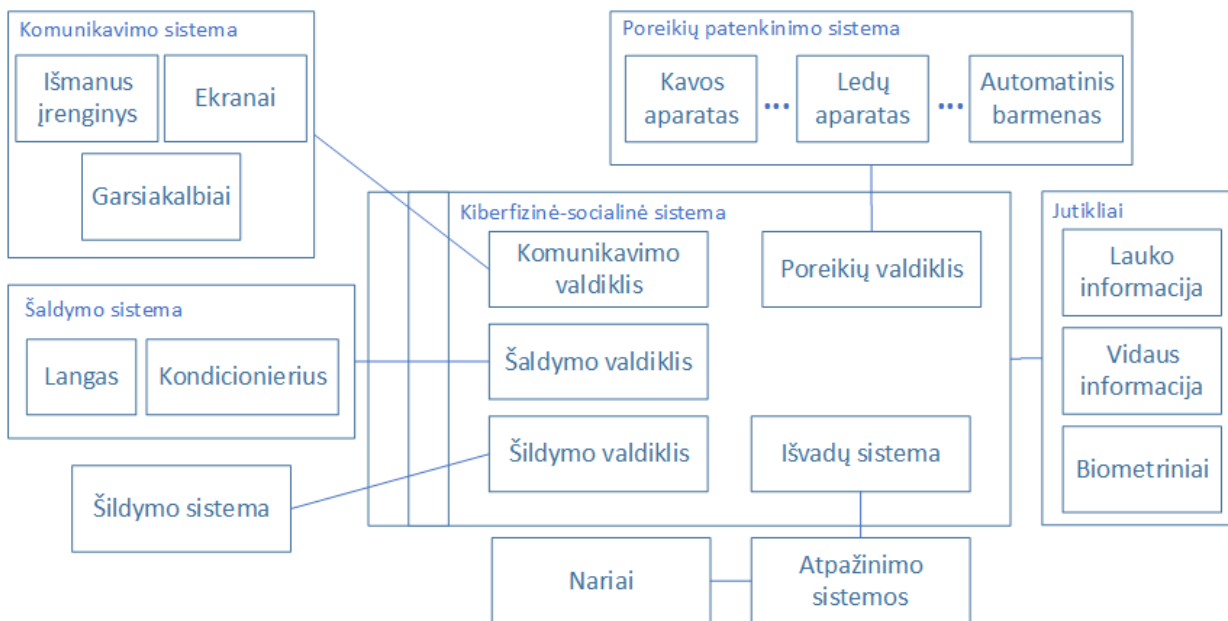
Kiberfizinės-socialinės sistemos sąvokų apmokymui reikalinga išvadų sistema (žr. 33 pav.). Realūs duomenys apie dalykinę sritį gaunami jutikliais, o nariai identifikuojami atpažinimo sistemos dėka. Bet koks atliktas sistemos veiksmas yra saugomas atliktų veiksmų archyve. Taip pat jai reikalingas socialiniuose portaluose atliktų veiksmų sąrašas. Turėdama šiuos duomenis sistema gali analizuoti žmonių poreikius, o anksčiau aprašytos sistemos pagal šias išvadas pritaikyti fizinį pasaulį.



33 pav. Probleminis freimas: nuolatinio informavimo išvadų sistema

Apibūdinus visus probleminius freimus, galima pereiti prie sistemos detalesnio aprašymo. Sistemai aprašyti reikalinga atnaujinta konteksto diagrama (žr. 34 pav.), kurioje yra nurodyti probleminiuose freimuose minimi sistemų pavadinimai. Tokia diagramų sąveika pagerina

reikalavimų atsekamumą ir padeda užtikrinti, kad visuose lygmenyse būtų kalbama apie vieną ir tą pačią sistemą.



34 pav. Konteksto diagrama loginių vienetų lygyje

Kaip jau minėjome, analizės rezultatų nusakymui gali būti taikomi įvairūs metodiniai karkasai, tačiau jie nėra pilnai tam tinkami. Mūsų nuomone, pritaikytų probleminių freimų naudojimas sistemos lygmenyje, turės įtakos žemesnių lygmenų gautų rezultatų detalumui, nes į analizę nuo pat pradžių yra įtraukiamas fizinis lygmuo.

Verta paminėti, kad šiame skyriuje esančios diagramos ir jų aprašai yra tik vienas iš galimų metodikos pritaikymo pavyzdžių, nes jos aprašymo detalumas priklauso nuo sistemos kūrėjo poreikio.

## REZULTATAI IR IŠVADOS

Šiame darbe buvo įgyvendintas iškeltas tikslas – probleminę domeno analizę pritaikyti kiberfizinių-socialinių sistemų reikalavimų inžinerijos dalykinės srities analizei ir modeliavimui.

### Darbo rezultatai

- Nustatytos ir apibrėžtos kiberfizinių ir kiberfizinių-socialinių sistemų, kaip sistemų tipų, savybės;
- Atsižvelgiant į teorinėje dalyje aprašytas KFSS savybes bei specifikacijų trūkumus, magistriniame darbe buvo pateiktas pasiūlymas, kaip modifikuoti klasikinius probleminius freimus atliekant kiberfizinių-socialinių sistemų reikalavimų analizę;
- Pasiūlytas metodinis karkasas, kuriame nurodyta iš kokių požiūrio taškų nagrinėjama sistema, kokios ir koku detalumu turi būti aprašytos jos savybės kuriant KFSS;
- Pasiūlytas metodinio karkaso egzempliorius, apibrėžiantis kiberfizinių-socialinių sistemų savybių modeliavimą. Dalykinės srities analizei atlikti pasirinkti probleminiai freimai, kurie praplėsti kiberfiziniam lygmeniui modeliuoti būtina kalba SysML ir komponentų komunikavimui skirta kalba FIPA-ACL;
- Pateikta metodikos realizacija konkrečiam atvejui.

### Darbo išvados

1. Reikalavimų inžinerijoje pagrindinis dėmesys paprastai yra skiriamas kibernetinei sistemos daliai ir jos programinės įrangos sukūrimui. Kiberfizinės-socialinės sistemos yra viena iš šiuolaikinių sistemų klasių. KFSS yra sudėtingos heterogeninės sistemos, kurių savarankiški komponentai tarpusavyje yra susieti daugeliu skirtingų ryšių. Šie komponentai turi gebėti adaptuotis prie pastoviai kintančio konteksto ir demonstruoti reikiamą susiformuojančią elgseną.
2. Sprendiniu, agentais grindžiamos ir kitos darbe apžvelgtos metodikos netinka specifikavimui dėl to, kad jomis nėra užtikrinamos reikiamos KFSS savybės, pavyzdžiui, tokios kaip prisitaikanti elgsena, pilnai ar dalinai autonomiškos posistemės, dinamiškas sistemos perkonfigūravimas skirtingose laiko skalėse.
3. Kiberfizinių-socialinių sistemų analizei ir modeliavimui turi būti naudojama probleminio domeno analizės metodika, kurioje, skirtingai nuo kitų, yra taikomas turinio atskyrimo principas. Dėl to yra gaunami du aprašymai: pirmasis pateikia informaciją apie dalykinę sritį ir joje esančias problemas, antrasis – kuriamą sistemą ir jos reikalavimus. Taip pat ši metodika neriboja projektavimo ir įgyvendinimų paradigimų pasirinkimo.
4. Norint pritaikyti probleminio domeno analizę, ją reikia papildyti atitinkamais elementais:

- a. Visuose probleminiuose freimuose turi atsirasti konteksto esybė;
- b. Valdamos sistemos probleminiuose freimuose valdoma esybė gali būti dviejų tipų (įprastinė ir sudėtinė);
- c. Informavimo sistemos probleminiuose freimuose kuriama mašina turi gebėti mokytis iš savo kuriamų rezultatų.

## LITERATŪROS SĄRAŠAS

- [Ars04] A. Arsanjani, „Service-oriented modeling and architecture: how to identify, specify, and realize your SOA“. Whitepaper, IBM Corporation, 2004.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://www.ibm.com/developerworks/library/ws-soa-design1/ws-soa-design1-pdf.pdf>>
- [BCG12] M. Broy, M. Victoria Cengarle, E. Geisberger. „Cyber-Physical Systems: Imminent Challenges“, 2012.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<[https://www.researchgate.net/publication/262160531\\_Cyber-Physical\\_Systems\\_Imminent\\_Challenges](https://www.researchgate.net/publication/262160531_Cyber-Physical_Systems_Imminent_Challenges)>
- [BGi11] Radhakisan Baheti ir Helen Gill. „Cyber-physical Systems“, 2011.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<http://ieeecs.org/sites/ieeecs/files/2019-07/IoCT-Part3-02CyberphysicalSystems.pdf>>
- [Bra02] I. K. Bray. „An introduction to requirements engineering“. Addison Wesley, 2002, 408 p.
- [DLV11] P. Derler, E. A. Lee, A. S. Vincentelli. „Modeling Cyber-Physical Systems“, 2011.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/5995279/>>
- [ELM+11] J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia, J. Zou. „Distributed Real-Time Software for Cyber-Physical Systems“, 2011.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/5995282/>>
- [EPR+14] S. Engell, R. Paulen, M. A. Reniers, C. Sonntag, H. Thompson. „Core Research and Innovation Areas in Cyber-physical Systems of Systems“, 2014.
- [GYa13] T. Guan, G. Yang. „Integration-Oriented Modeling of Cyber-physical Interactive Process“, 2013.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/6682275/>>
- [GPG+14] V. Gunes, S. Peter, T. Givargis, F. Vahid. „A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems“, 2014.
- [Hor14] I. Horvath. „What the Design Theory of Social-Cyber-Physical Systems Must Describe, Explain and Predict?“. An Anthology of Theories and Models of Design, Springer, 2014, 99 – 120 p.
- [HSG16] J. Huang, M. D. Seck, A. Gheorghe. „Towards Trustworthy Smart Cyber-Physical-Social Systems in The Era of Internet of Things“, 2016.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/7542961/>>
- [Jac01] M. Jackson. „Problem Frames: Analysing and Structuring Software Development Problems“. Addison-Wesley, 2001.
- [Jac96] I. Jacobson. „Object oriented software engineering: a use-case driven approach“. Addison Wesley, 1996.
- [Kim14] Junsung Kim. „Dependable Cyber-Physical Systems“, 2014.  
[žiūrėta 2020-05-26]. Prieiga per internetą:



<<https://www.semanticscholar.org/paper/Dependable-Cyber-Physical-Systems-Kim/497083d89b6bab80e0a9c94a965f3d12d32f4878>>

- [KMP+06] Matthias Kirchmayr, Mark Müller, Birgit Penzenstadler, Ernst Sikora, Thorsten Weyer. „Requirements Engineering und Management softwareintensiver Eingebetteter Systeme“ 2006.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<[https://remses.eu/remses.org/download/REMsES\\_Essenzieller-Leitfaden.pdf](https://remses.eu/remses.org/download/REMsES_Essenzieller-Leitfaden.pdf)>
- [LYW+11] Z. Liu, D. Yang, D. Wen, W. Zhang, W. Mao. „Cyber-Physical-Social Systems for Command and Control“, 2011.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/5968110>>
- [Lup07] A. Lupeikienė, „Teoriniai ir technologiniai informacinių sistemų aspektai“. Vilnius, 2007.
- [LZZ13] C. Liu, W. Zhang, H. Zha, Z. Jin. „Analyzing Early Requirements of Cyber-Physical Systems Through Structure and Goal Modeling“, 2013.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/6805400>>
- [NIST13] National Institute of Standards and Technology. „Foundations for Innovation in Cyber Physical Systems“. Workshop Report, 2013.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://www.nist.gov/system/files/documents/el/CPS-WorkshopReport-1-30-13-Final.pdf>>
- [NSF13] National Science Foundation, „Cyber Physical Systems NSF10515“, 2013.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm>>
- [NSF17] National Science Foundation. „Cyber-Physical Systems (CPS) (nfs17529)“, 2017.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://www.nsf.gov/pubs/2017/nsf17529/nsf17529.pdf>>
- [Poh10] K. Pohl, „Requirements Engineering“, 2010.
- [Poo10] R. Poovendran. „Cyber–Physical Systems: Close Encounters Between Two Parallel Worlds“, 2010.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/5512708>>
- [RON17] H. Ramadhan, D. Oktaria, G. B. B. Nugraha. „Road Traffic Signal Control using Cyber Physical Social System“, 2017.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/8267947>>
- [RRW12] J. O. Ringert, B. Rumpe, A. Wortmann. „A Requirements Modeling Language for the Component Behavior of Cyber Physical Robotics Systems“, 2012.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://arxiv.org/ftp/arxiv/papers/1409/1409.0394.pdf>>
- [SAH13] A. Sheth, P. Anantharam, C. Henson. „Physical-Cyber-Social Computing: An Early 21st Century Approach“. IEEE Intelligent Systems, 2013, 78 – 82 p.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/6468021>>
- [SCC17] V. Seidita, A. Chella, M. Carta. „Capturing citizens — Emerging needs: Using social networks in smart cities“, 2017.

- [žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/8240532>>
- [Sch14] Bernhard Schätz. „The Role of Models in Engineering of Cyber-Physical Systems – Challenges and Possibilities“, 2014.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://pdfs.semanticscholar.org/a5ea/446c8b75a103e72dca25ee86b44b9b591314.pdf>>
- [Wan10] F. Wang. „The Emergence of Intelligent Enterprises: From CPS to CPSS“, 2010.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/abstract/document/5552591>>
- [WCC+17] G. Wang, X. Cai, Z. Cui, G. Min, J. Chen. „High Performance Computing for Cyber Physical Social Systems by Using Evolutionary Multi-Objective Optimization Algorithm“, 2017.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/7927724>>
- [WTN06] R. Wirfs-Brock, P. Taylor, J. Noble. „Problem Frame Patterns“, 2006.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<http://www.wirfs-brock.com/PDFs/ProblemFramePatterns.pdf>>
- [WZY+17] S. Wang, A. Zhou, M. Yang, L. Sun, C. Hsu, F. Yang. „Service Composition in Cyber-Physical-Social System“, 2017.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/7865933>>
- [XMa11] F. Xia, J. Ma. „Building Smart Communities with Cyber-Physical Systems“, 2011.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://arxiv.org/ftp/arxiv/papers/1201/1201.0216.pdf>>
- [Zha13] L. Zhang. „Requirement Specification for Transportation Cyber Physical Systems“, 2013.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/6682274>>
- [ZYL+16] J. Zeng, L. T. Yang, M. Lin, H. Ning, J. Ma. „A survey: Cyber-physical-social systems and their system-level design methodology“, 2016.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://www.sciencedirect.com/science/article/abs/pii/S0167739X1630228X>>
- [ZTL+16] Y. Zhu, Y. Tan, R. Li, X. Luo. „Cyber-physical-social-thinking modeling and computing for geological information service system“. International Journal of Distributed Sensor Networks, volume 12, issue 11, 2016.  
[žiūrėta 2020-05-26]. Prieiga per internetą:  
<<https://ieeexplore.ieee.org/document/7428352>>