

KAUNO TECHNOLOGIJOS UNIVERSITETAS
VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

NERIJUS JUSAS

**Požymių modeliais grindžiamas daiktų interneto
sistemų kūrimas**

Daktaro disertacijos santrauka
Technologijos mokslai, informatikos inžinerija (07T)

2017, Kaunas

Disertacija rengta 2012-2017 metais Kauno technologijos universiteto informatikos fakultete kompiuterių katedroje.

Mokslinis vadovas:

Prof. dr. Algimantas Venčkauskas (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Redagavo: Aurelija Gražina Rukšaitė (leidykla „Technologija“)

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:

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

Prof. habil. dr. Gintautas DZEMYDA (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T);

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

Doc. dr. Raimundas MATULEVIČIUS (Tartu universitetas, Estija, technologijos mokslai, informatikos inžinerija – 07T);

Doc. dr. Tomas SKERSYS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija bus ginama viešame informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2017 m. birželio 23 d. 14 val. Kauno technologijos universiteto disertacijų gynimo salėje.

Adresas: K. Donelaičio g. 73-403, 44249 Kaunas, Lietuva. Tel. (370) 37 300 042; faks. (370) 37 324 144; el. paštas doktorantura@ktu.lt.

Disertacijos santrauka išsiųsta 2017 m. gegužės 23 d. Su disertacija galima susipažinti internetinėje svetainėje <http://ktu.edu>, Kauno technologijos universiteto (K. Donelaičio g. 20, 44239 Kaunas) ir Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, LT-10223 Vilnius) bibliotekose.

1. IŽANGA

Netolimoje ateityje ne tik žmonės ir kompiuteriai, bet ir kasdieninio gyvenimo daiktai bus sujungti tarpusavyje, taip sukuriant naują kompiuterinę infrastruktūrą – daiktų internetą (DI). Pagrindinis DI tikslas – užtikrinti, kad daiktai būtų sujungti bet kada, bet kur, su bet kuo ir su kiekvienu, naudojant bet kokį kelią/tinklą ir bet kokią paslaugą. Semantiškai DI reiškia naują, visą pasaulį apimantį labai heterogenišką tinklą, kuris jungia unikaliai adresuojamus objektus ir yra pagrįstas standartiniais komunikacijos protokolais.

Pagrindiniai DI sistemų (*angl. IoT application*) ribojimai yra saugumas/privatumas, energijos vartojimas ir aplinkos veiksniai. Pirmasis veiksnys atsiranda dėl tikimybės, kad informacija bus perimta ir pakeista apdorojimo arba siuntimo metu. Antrasis – dėl to, kad naudojami prietaisai su autonominiais energijos šaltiniais (kraunamais akumulatoriais). Trečiasis – dėl naudojamų technologijų, komunikacijų protokolų įvairumo, aplinkos radijo triukšmų, kurie trikdo duomenų siuntimą. Paminėti veiksniai yra artimai susiję ir itin sudėtingi savaip. Visi trys veiksniai kartu apibrėžia DI sistemos (DIS) paslaugų kokybę. Todėl paslaugų kokybė turėtų atsirasti kaip pagrindiniai nefunkciniai reikalavimai kuriant DI sistemas. DIS reikalavimai paslaugų kokybei yra labai skirtingi, net jeigu sprendimai yra labai panašūs.

DIS susideda iš mazgų (jutiklis, valdiklis) ir programinės įrangos, kuri surenka duomenis ir valdo šiuos mazgus. Mazgų skaičius gali būti nuo keleto iki kelių šimtų ar daugiau, jų skaičius priklauso nuo DIS. Norint užtikrinti skirtingas DIS funkcijas, mazgai gali būti sujungiami į grupes. DIS naudojamų mazgų veikimas ir struktūra gali smarkiai skirtis, priklausomai nuo taikymų reikalavimų. Iš kitos pusės, gali būti naudojami tokie patys mazgai, kurie naudoja tą pačią programinę įrangą, bet veikia skirtingais režimais, taip užtikrindami reikiamą DIS veikimą. Nuolatos kylant sistemų sudėtingumui, sistemų kūrimo metodai pradėjo remtis pramonėje sėkmingai naudojamais metodais, tokiais kaip produktų šeimos (PŠ) kūrimas (*angl. Product lines*), modeliais grindžiamas sistemų kūrimas ar modeliais grindžiamas sistemų produktų šeimos kūrimas. Šie metodai yra orientuoti į pakartotinį aparatinių ir programinių komponentų panaudojimą. Tai užtikrina greitesnį projekto vystymą ir sumažina galimų klaidų skaičių galutiniame produkte.

Dėl DIS naujumo ir specifikos produktų šeimos kūrimo metodologijos nėra plačiai naudojamos, norint užtikrinti geresnę DIS kokybę, didesnę produktyvumą ir pakartotinį komponentų panaudojimą, nepaisant to, kad DIS galima suskaidyti į problemines sritis, kur sistemos,

esančios toje pačioje srityje, turi daug bendrų savybių.

Šios disertacijos tikslas – sukurti ir ištirti DIS produktų šeimos kūrimo metodą. Šiuo metu vienos iš pagrindinių DIS kūrimo problemų yra nepakankamas kūrimo procesų automatizavimo lygis ir pakartotinis esančių sprendimų panaudojimas, integracija ir pritaikymas.

1.1. Darbo objektas

Šio darbo objektas yra požymių modeliais pagrįstas metodas, skirtas kurti DI sistemų produktų šeimas.

1.2. Darbo tikslas

Šio darbo tikslas yra DI sistemų kūrimas probleminėje srityje naudojant produktų šeimos metodologijas, atsižvelgiant į DI sistemų kūrimo iššūkius: reikalavimų kompleksiskumas (saugumas, energijos vartojimas), technologijų heterogeniškumas ir aplinkos faktoriai.

1.3. Darbo uždaviniai

1. Išanalizuoti tipinius DI sistemų kūrimo reikalavimus ir iššūkius.
2. Sukurti DI sistemų kūrimo metodą, pagrįstą sistemų produktų šeimos kūrimo principais.
3. Pasiūlyti požymių modelių ir specializuotų konfigūracijų transformavimo į DI sistemos karkasą taisykles.
4. Įgyvendinti DI sistemos prototipą, naudojant pasiūlytą DI sistemų kūrimo metodą.

1.4. Mokslinis naujumas

Darbe pasiekti šie nauji rezultatai:

1. Pasiūlytas naujas požymių modeliais pagrįstas metodas, skirtas kurti DI sistemų produktų šeimas.
2. Pasiūlyti funkcinių ir nefunkcinių reikalavimų požymių modeliai, kurie atvaizduoja DI sistemos reikalavimų variantiškumą (produktų šeimas) probleminėje DI srityje ankstyvoje sistemos kūrimo stadijoje.
3. Pateiktos bendrinių DI srities požymių modelių ir konkrečios DI sistemos reikalavimų suderinimo taisyklės.
4. Pasiūlytas DI sistemos karkaso generavimo iš požymių modelio būdas.

1.5. Darbo rezultatų praktinė vertė

DI sistemų kūrimas yra labai sudėtingas ir daug laiko reikalaujantis procesas, nes šio proceso metu reikia suderinti programinę įrangą, techninę įrangą ir techninės įrangos veikimo režimus, kad būtų užtikrintas kiek įmanoma geresnis DI sistemos veikimas (paslaugų kokybė). Be to, DI sistemoms yra būdingas suskaidymas į sritis, kuriose jos turi nemažai bendrų savybių, todėl DI sistemoms kurti gali būti naudojamos produktų šeimos kūrimo metodologijos. Šios metodologijos leidžia jau sukurtas sistemas komponentus panaudoti kuriant kitas sistemas. Atsižvelgiant į tai, buvo pasiūlytas DI sistemų kūrimo metodas, kuris:

- pateikia galimus DI sistemos įgyvendinimo variantus (produktų šeimas);
- iš galimų įgyvendinimo variantų parenka vieną variantą, kuris geriausiai atitinka kuriamos DI sistemos reikalavimus;
- iš parinkto geriausio varianto sugeneruoja DI sistemos karkasą.

Darbe pateikiami rezultatai buvo panaudoti vykdant projektą „Daiktų ir paslaugų interneto technologijų ir jų infrastruktūros išmaniosioms aplinkoms tyrimas bei kūrimas“, Nr. VP1-3.1-ŠMM-08-K-01-018.

1.6. Ginami teiginiai

1. Požymių modeliais pagrįstas DI sistemų kūrimo metodas yra skirtas kurti DI sistemų produktų šeimas tam tikroje DI srityje.
2. Bendriniai požymių modeliai aprašo analizuojamos DI sistemų srities variantiškumą, kompleksiškumą, technologijų ir aplinkos heterogeniškumą.
3. Bendrinių modelių transformacijos rezultatas yra DI sistemos Pareto optimalus modelis, iš kurio yra generuojamas DI sistemos karkasas.

1.7. Mokslinis pripažinimas

Darbo tema paskelbta 6 straipsniai, iš jų 4 straipsniai žurnaluose, įtrauktuose į „ISI Web of Science“ su citavimo indeksu, 1 straipsnis konferencijų pranešimų medžiagoje. Taip pat 1 straipsnis kituose recenzuojamuose leidiniuose.

Tyrimų rezultatai pristatyti užsienyje ir Lietuvoje vykusiose tarptautinėse konferencijose:

1. 18th International conference Electronics 2014, Palanga;
2. 20th International conference on information and software technologies, ICIST 2014, Druskininkai;
3. 12th Annual International Conference on Information Technology & Computer Science 2016, Athens.

1.8. Disertacijos struktūra

Disertacija yra sudaryta iš penkių skyrių, priedų ir literatūros sąrašo.

Pirmame disertacijos skyriuje aprašomas tyrimo aktualumas, tyrimo objektas, sprendžiama problema, darbo tikslai ir uždaviniai, pateikiami ginami teiginiai, praktinė pasiektų rezultatų reikšmė ir disertacijos struktūra.

Antrasis skyrius yra apžvalginis, jame analizuojamas daiktų internetas. Apžvalga yra orientuota į DIS, DIS kūrimo iššūkius, DIS paslaugų kokybės užtikrinimą ir metodus, skirtus DIS kurti. Šio skyriaus tikslas – supažindinti skaitytoją su disertacijoje vartojama terminologija.

Trečiame skyriuje aprašomas pasiūlytas DI sistemų kūrimo metodas, pagrįstas pažymių modeliais. Skyriaus pradžioje yra aprašomas pasiūlytas metodas. Toliau yra aprašomos pasiūlyto metodo penkios fazės: modelių kūrimo, agregavimo, specializavimo, generavimo ir įgyvendinimo, plačiau aprašomi kiekvienoje fazėje atliekami veiksmai.

Ketvirtame skyriuje aprašomas DIS prototipo įgyvendinimas naudojant pasiūlytą DI sistemų kūrimo metodą. Metodas buvo taikomas žmogaus sveikatos parametrų stebėjimo DI sistemai sukurti.

Penktime skyriuje yra pateikiamos galutinės darbo išvados.

2. DAIKTŲ INTERNETAS

Terminas daiktų internetas (DI) pirmą kartą buvo paminėtas Peter T. Lewis [Saha et al., 2017] 1985 metais. Tuo metu šis terminas plačiai neišplito, tačiau šiandien DI aprėpia daugumą gyvenimo aspektų nuo tarpusavyje sujungtų namų ir miestų iki tarpusavyje sujungtų automobilių ir kelių. Pagrindinis DI tikslas – užtikrinti, kad daiktai būtų sujungti bet kada, bet kur, su bet kuo, panaudojant bet kokius internetinius kelius ir paslaugas [Smith, 2012]. Semantiškai DI reiškia naują, labai heterogenišką, visa pasaulį apimančią tinklą, kuris apima unikalios adresuojamus objektus, tam naudodamas standartinius interneto ryšio protokolus [Labiod et al., 2007].

Yra keletas DI apibrėžimų. Mūsų manymu, DI apibrėžimas, pateiktas Gubbi ir kitų [Gubbi et al., 2013], geriausiai atitinka DI sąvoką šioje disertacijoje. DI yra suprantamas kaip išmaniosios aplinkos karkasas, kuris apima jutiklius ir valdiklius. Toks komponentų sujungimas leidžia dalintis informacija tarp skirtingų platformų, naudojant tą patį karkasą. Toks karkasas naudoja interneto technologijas sugeneruotiems duomenims apdoroti, perduoti, saugoti ir atvaizduoti.

„Daiktai“ DI – tai fiziniai objektai, kurie sąveikauja tarpusavyje, naudodami interneto ryšio protokolus. Šiame kontekste „daiktai“ (fiziniai objektai) – tai elektroniniai prietaisai, kurie gali būti sujungti interneto priemonėmis, taip pat kiti objektai, neturintys ryšio galimybių. Objektai su ryšio galimybėmis gali sąveikauti su kitais objektais naudojant bendrus ryšio protokolus, dažniausiai naudojami belaidžio ryšio protokolai. Objektai, neturintys ryšio galimybių, gali būti naudojami parodyti informaciją apie objektus arba keisti jų būsenas.

Atzori ir kt. [Atzori et al., 2010] pateikė technologijų, kurios dažniausiai siejamos su DI sistemomis, apžvalgą. Autoriai išskyrė tris pagrindines technologijų kategorijas, kurios yra DIS pagrindas: daiktai, ryšio tinklas ir semantika. Taip pat šiose apžvalgose yra teigiama, kad DI sistemoms būdingos bendros savybės: technologijos, ryšio protokolai, duomenų persiuntimas ir paslaugų kokybė (*angl. quality of service*).

2.1. DI sistemos

Naudojant DI technologijas, galima sukurti beveik begalinį skaičių įvairiausių sistemų. IERC [IERC, 2009] organizacija nustatė ir aprašė pagrindines DI sistemų sritis, kuriose gali būti įgyvendinami DI sprendimai: išmanūs miestai, išmanusis transportas, išmanūs pastatai, išmanioji energija, išmanioji pramonė, išmanioji sveikatos apsauga. Chen ir kiti [Chen et al., 2014] išskyrė devynias pagrindines DI sistemų sritis, kurios yra pašios į IERC išskirtas. DI sistemų sričių skaičius gali būti ir kitas, tačiau daugumą DI sistemų galima priskirti kuriai nors iš pateiktų sričių. Tai rodo, kad DI sistemą galima suskirstyti į sritis, kurios turi daug bendrų savybių.

Kuriant DIS reikia įvertinti šias būdingas DI sistemų charakteristikas [Patel et al., 2011]:

- prietaisų heterogeniškumas – dažniausiai DI sistemos turi sujungti įvairius jutiklius ir valdiklius ir užtikrinti duomenų mainus tarp jų;
- įgyvendinimo platformų heterogeniškumas – DI sistemose skirtingi prietaisai gali būti įgyvendinti naudojant įvairias aparatines ir programines platformas: Arduin, .Net Gadgeteer ir kitas;

- DI sistemų dydis – prietaisų skaičius DI sistemoje gali būti nuo kelių šimtų iki kelių milijonų, prietaisai atlieka skirtingus veiksmus;
- DIS panašumas srityje – skirtingos DI sistemos turi nemažai bendrų savybių, kurios būdingos tam tikrai DI sričiai.

2.2. Komunikacija tarp „daiktų“ daiktų internete

Dauguma kompiuterių tinklų naudoja TCP/IP protokolus ir laidinius arba belaidžius komunikacijos tinklus. Nepaisant komunikacijos protokolų įvairumo, TCP/IP protokolai yra faktinis standartas dabartiniame internete. DI sistemose objektai gali komunikuoti skirtingais lygiais, naudodami skirtingus protokolus. Yra sukurta nemažai komunikacijos protokolų, pagrįstų esamais protokolais ir skirtų DIS įgyvendinti.

Nepaisant naujų pasiūlymų, dauguma DIS naudoja egzistuojančius protokolus, veikiančius TCP/IP principu. Dažniausiai DIS naudoja standartinius belaidžius protokolus: Bluetooth, Bluetooth low energy, Wi-Fi, ZigBee ir kt. Šie protokolai yra skirtingi vertinant jų energijos sąnaudas, veikimo režimus, komunikacijos ir saugumo galimybes. Išvardinti protokolų skirtumai turi įtakos DIS paslaugų kokybei. Todėl kuriant DIS reikia įvertinti šias protokolų savybes, norint parinkti DIS komunikacijos protokolą (-us), kuris (-ie) geriausiai atitinka DIS keliamus paslaugų kokybės reikalavimus.

2.3. Energijos sąnaudų problemos DI sistemose

Energijos sąnaudos yra viena iš didžiausių DI problemų. Komunikacija yra viena iš daugiausiai energijos reikalaujančių DI užduočių. Energijos sąnaudoms sumažinti yra sukurti komunikacijos protokolai, kurie vartoja mažiau energijos (ZigBee, RFID ir kiti). Tačiau komunikacijos charakteristikos DI skiriasi nuo tradicinių belaidžių tinklų, nes dauguma prietaisų naudoja baterijas kaip energijos šaltinį, taip pat kai kada prietaisų, dalyvaujančių komunikacijoje, skaičius yra labai didelis. Be to, dažniausiai DIS tinklas turi veikti ilgą laiką be žmonių įsikišimo. Nors sukurta naujų komunikacijos protokolų, DI sprendimai dažniausiai naudoja standartinius protokolus, todėl iškyla problema, kaip šiuos protokolus įvertinti pagal DIS reikalavimus ir pasiūlyti tinkamiausią komunikacijos protokolą (-us) DIS įgyvendinti

2.4. Saugumo ir privatumo problemos DI sistemose

Dauguma DI prietaisų turi ribotus energijos ir skaičiavimo išteklius, todėl saugumas yra didelė DIS problema. Dėl minėtų priežasčių dauguma esamų kriptografinių algoritmų ir protokolų negali būti naudojami kuriant DIS. DI heterogeniškumas taip pat turi įtakos saugumo

reikalavimams, nes skirtingi komunikacijos protokolai ir algoritmai turi būti sujungti, norit užtikrinti DIS duomenų saugumą. Todėl reikia analizuoti komunikacijos protokolus ir jų teikiamas saugumo galimybes, nes kol kas tai yra vienas iš pagrindinių DI sprendimų duomenų saugumo užtikrinimo būdų. Sudėtingesni protokolai užtikrina geresnes saugumo galimybes, tačiau reikalauja daugiau energijos jiems įgyvendinti, o tai kertasi su ribotų energijos išteklių problema. Todėl, kuriant DIS, komunikacijos protokolų teikiamo saugumo ir energijos vartojimo problemas reikia analizuoti ir spręsti kartu.

2.5. Paslaugų kokybė DI sistemose

DIS paslaugų kokybė yra aprašoma taip: „DIS paslauga užtikrina su daiktu susijusias paslaugas, kurios atsižvelgia į daikto ribojimus, tokius kaip privatumo užtikrinimas ir semantinio vientisumo užtikrinimas tarp daikto ir susietų virtualių daiktų“ [Vermesan and Friess, 2014]. Paslauga – tai tam tikras funkcijų rinkinys, kuris yra apibrėžiamas paslaugų kokybe, susidedančia iš saugumo, energijos vartojimo, vartotojo pasitenkinimo ir patikimumo komponentų.

Kadangi paslaugų kokybę sudaro įvairiausi komponentai, turintys savo parametrus ir atributus, kurie daro įtaką paslaugų kokybei, būtina šiuos skirtingus parametrus įvertinti kiekvienos DIS atveju. Skirtingiems parametrų įvertinti yra naudojami optimizavimo metodai [Shaoshuai et al., 2011, Li et al., 2014], kurių tikslas – rasti geriausių parametrų derinį pagal kuriamos DIS reikalavimus, pvz.: komunikacijos protokolai turi užtikrinti kiek įmanoma didesnę siunčiamų duomenų saugumą, suvartojant kiek įmanoma mažiau energijos.

2.6. DI sistemų kūrimo metodai

DI sistemų kūrimas yra sudėtingas, nes jos sudarytos iš daugybės komponentų, tokių kaip jutikliai, valdikliai, komunikacijos protokolai, programinė įranga ir kita. Todėl, kuriant DIS, šie komponentai turi būti sujungti. Norint tai atlikti, reikia sujungti tiek programinės įrangos, tiek aparatinės įrangos sritis, kurios yra naudojamos kuriant DIS. Šiam darbui palengvinti yra naudojamos modeliais grindžiamos arba produktų šeimos kūrimo metodologijos. Kuriant DIS modeliais grindžiamomis metodologijomis, pirmiausia yra sukuriamas DIS modelis, iš kurio vėliau yra sugeneruojamas sistemos programinis kodas. DIS modeliai yra kuriami naudojant modeliavimo kalbas, kurios gali būti tiek bendros paskirties, pvz., UML, tiek skirtos tik DIS modeliuoti: Midgar [García et al., 2014], DSL-4-IoT [Salihbegovic et al., 2015], Node-RED DSL [Kawamoto, 2016]. Taip pat yra specifinių modeliais grindžiamų

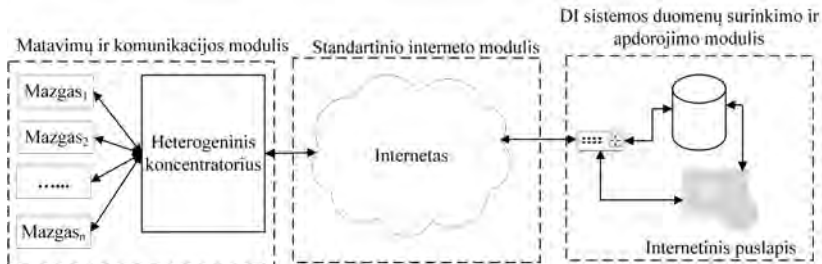
metodologijų, kurios yra skirtos tik DIS kurti, šios metodologijos aprašo visus veiksmus – nuo DIS modelio sukūrimo iki programinio kodo generavimo. Riedel ir kt. [Riedel et al., 2010] aprašytas metodas, skirtas tinklo objektams ir organizacijos sistemoms sujungti. Grace ir kt. [Grace et al., 2016] ir Patel ir Cassou [Patel and Cassou, 2015] pateikti metodai skirti aprašyti DIS prietaisų ir komunikacijos protokolų heterogeniškumą. Kaip matyti, pasiūlyti modeliais grindžiami DIS kūrimo metodai skirti modeliuoti tik vieną sistemą iš karto. Tačiau dauguma dabartinių DIS sistemų yra orientuotos į taikymų sritis, kuriose jos turi daug bendrų savybių. Šie faktai rodo, kad produktų šeimos kūrimo metodologijos gali būti naudojamos DIS kurti.

Produktų šeimos kūrimo metodologijų principas yra aprašyti galimus DIS kūrimo variantus (produktų šeimas) ankstyvoje sistemos kūrimo stadijoje, išskiriant komponentus, kurie yra bendri ir skirtingi visoms tos pačios srities sistemoms. Taip galima pagreitinti sistemų kūrimą ir pakartotinai panaudoti jau sukurtus komponentus. Norint išskirti komponentus, kurie yra bendri ir skirtingi tos pačios srities sistemoms, yra naudojamos variantiškumo aprašymo kalbos, pvz., požymių modeliai, specialios UML notacijos ir kitos. Nepaisant daugybės įvairiausių variantiškumo aprašymo kalbų, požymių modeliai yra populiariausia variantiškumo aprašymo kalba tiek pramonėje, tiek mokslo institucijose, kuriant ir plečiant sistemas įvairiose srityse [Berger et al., 2013]. Tai rodo, kad požymių modeliai yra nepriklausomi nuo taikymo srities. DIS kurti yra pasiūlyta metodų, kurie naudoja produktų šeimos metodologijas. Anon ir kt. [Anon et al., 2014] aprašytas daugiasluoksnis DIS kūrimo karkasas, kuris naudoja variantiškumą (produktų šeimas) kuriant debesų technologijomis pagrįstas DIS. Ayala ir kt. [Ayala et al., 2015] pasiūlė metodą, naudojantį produktų šeimos metodologijas kuriant daugiaagentes DIS. Conejar ir Kim [Conejar and Kim, 2016] pasiūlė saugių DIS kūrimo metodą, skirtą mobiliems telefonams, kur produktų šeimos metodologijos skirtos aprašyti variantiškumą trimis lygiais: prieiga, sistema ir technologijos.

Nė vienas iš analizuotų DIS kūrimo metodų nevertino saugumo reikalavimų, energijos ir skaičiavimo išteklių ribotumo, technologijų įvairumo ar šių technologijų veikimo režimų įtakos DIS paslaugų kokybei apskritai, nors DIS analizė parodė, kad šie veiksniai yra svarbūs DIS kūrimui ir teikiamų paslaugų kokybei.

3. PASIŪLYTAS DI SISTEMŲ KŪRIMO METODAS

DIS analizė parodė, kad dauguma jų turi labai panašią struktūrą, kuri gali būti pritaikyta daugumai DIS. Principinė DIS struktūra yra sudaryta iš trijų pagrindinių modulių (žr. 3.1): matavimų ir komunikacijos (MKM), standartinio interneto (SIM) ir DIS duomenų surinkimo ir apdorojimo (DISDSAM) modulio.



3.1 pav.: DI sistemų principinė struktūra

MKM atlieka dvi pagrindines funkcijas: rinkti duomenis ir siųsti juos į DISDSAM arba atlikti veiksmus pagal valdymo signalus, gautus iš DISDSAM. MKM susideda iš mazgų tinklo ir heterogeninio koncentraatoriaus (HK). Kadangi mazgų tinklas komunikacijai dažniausiai naudoja mažo atstumo belaidžius komunikacijos protokolus, kurie tiesiogiai negali siųsti duomenų per SIM, todėl yra naudojamas HK, kuris surenka duomenis iš MKM ir juos apdoroja, kad būtų galima siųsti per SIM.

DISDSAM surenka, apdoroja ir saugo MKM surinktus duomenis, jeigu reikia po duomenų apdorojimo, siunčia valdymo signalus į MKM. Taip pat šis modulis gali būti atsakingas už surinktų duomenų atvaizdavimą vartotojui, naudojant įvairius grafinius paketus.

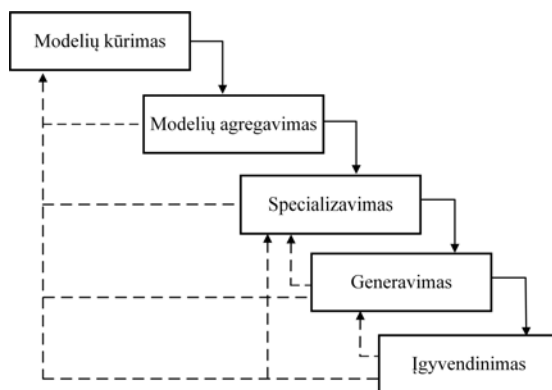
Tokia trijų modulių DIS struktūra leidžia panaudoti produktų šeimos metodologijas kuriant MKM, nes šis DIS modelis apima daug savybių, kurios yra bendros daugumai DIS toje pačioje srityje. Taip naudojant produktų šeimos metodologijas DIS kurti būtų užtikrintas aparatinių ir programinių komponentų pakartotinis panaudojimas, ir tai pagreitintų DIS kūrimą bei pagerintų jų kokybę.

Atsižvelgiant į tai ir į anksčiau aprašytas DIS kūrimo problemas, buvo pasiūlytas DIS kūrimo metodas, pagrįstas požymių modeliais, kuris ankstyvoje DIS kūrimo stadijoje aprašo galimas sistemos konfigūracijas (sistemos variantiškumą), įvertina šias konfigūracijas ir parenka tinkamiausią DIS įgyvendinti. DIS variantiškumui (produktų šeimai) anks-

tyvoje sistemos kūrimo stadijoje aprašyti buvo pasirinkta naudoti požymių modelius, nes: 1) šie modeliai yra plačiai naudojami variantiškumui aprašyti; 2) yra nepriklausomi nuo taikymo srities; 3) yra nemažai programinės įrangos, skirtos kurti požymių modelius. 3 ir 4 skyriuose DIS konfigūracija yra suprantama kaip sąrašas, kuris aprašo komponentų, komunikacijos protokolų, energijos sąnaudų ir saugumo parametrus.

3.1. DI sistemų kūrimo metodas

Atsižvelgiant į DIS kūrimo iššūkius, buvo pasiūlytas penkių fazių DI sistemų kūrimo metodas. Pirmos dvi metodo fazės yra skirtos DIS sričiai analizuoti ir DIS kūrimo konfigūracijų erdvei sudaryti. Likusios trys fazės skirtos konkrečiai DIS kurti atliekant tokius veiksmus: 1) parinkti tinkamiausią DIS kūrimo konfigūraciją iš galimos kūrimo erdvės; 2) generuoti DIS karkasą; 3) įgyvendinti DIS. Pasiūlyto metodo struktūra yra pavaizduota 3.2 pav. Tokia struktūra buvo pasirinkta siekiant supaprastinti DIS srities analizę ir užtikrinti geresnį komponentų pakartotinį panaudojimą kuriant skirtingas DIS. Siūlomas metodas yra skirtas kurti DIS matavimų ir komunikacijos modulius.



3.2 pav.: DIS kūrimo metodas

Siūlomas metodas susideda iš penkių fazių:

1. **Modelių kūrimas** — žinių apie probleminę DIS sritį surinkimas ir modelių sudarymas. Šiam tikslui įgyvendinti šioje fazėje yra sukuriami du modeliai: funkcinių ir nefunkcinių reikalavimų, aprašantys galimus DIS kūrimo variantus.
2. **Agregavimas** — agreguoto modelio sukūrimas, kuris yra gaunamas agreguojant sukurtus funkcinių ir nefunkcinių reikalavimų

modelius. Pagrindinis šio sluoksnio tikslas – užtikrinti kiek įmanoma didesnę pakartotinio panaudojimo potencialą, naudojant modeliais paremtas paradigmas. Modelių agregavimo taisyklės yra: a) modeliai turi būti tikslūs; b) modeliai turi būti sukurti naudojant požymių modelius; c) modeliai turi būti abstraktūs, kad būtų galima užtikrinti jų pakartotinį panaudojimą.

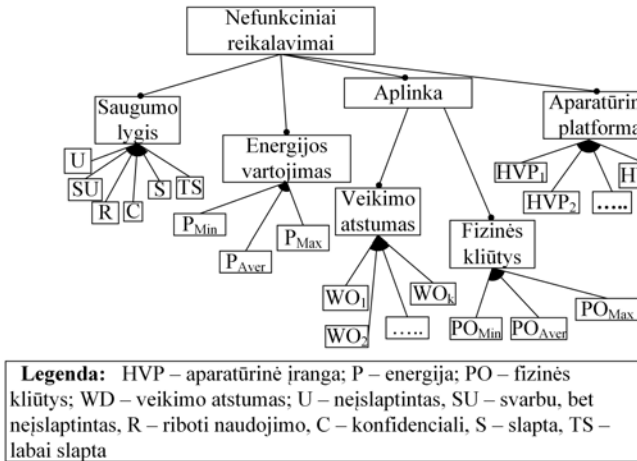
3. **Specializavimas** — skirtas abstrakčiam agreguotam modeliui transformuoti į modelį, kuris pateikia galimas DIS konfigūracijas (DIS sprendimų erdvę), atsižvelgiant į konkrečios sistemos reikalavimus. Atlieka šių konfigūracijų įvertinimą pagal kuriamos DIS paslaugų kokybės reikalavimus ir pateikia vieną specializuotą konfigūraciją, kuri geriausiai atitinka keliamus reikalavimus.
4. **Generavimas** — iš specializuotos DIS konfigūracijos generuojamas sistemos karkasas.
5. **Įgyvendinimas** — sugeneruotas DIS karkasas papildomas reikiama komponentais ir kodu, sukuriama veikianti sistema.

3.2. Modelių kūrimo fazė

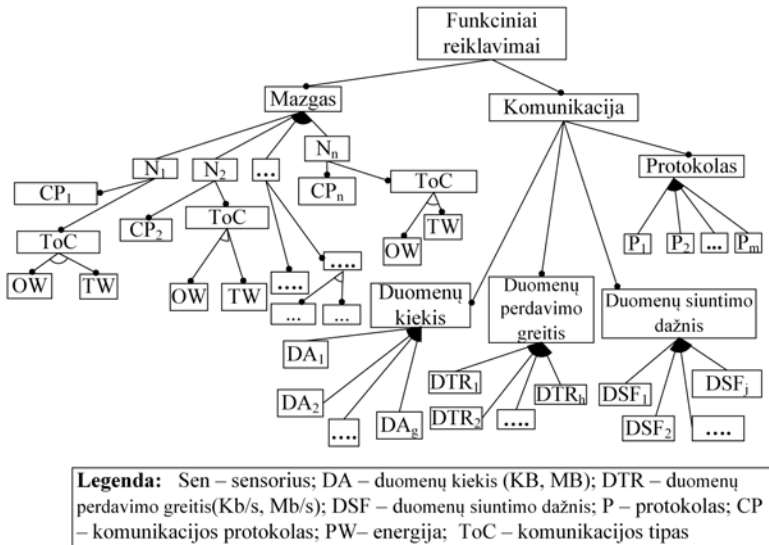
DIS modelių kūrimo fazėje analizuojama DIS probleminė sritis, sukuriama požymių modeliai funkciniais ir nefunkciniais reikalavimams. Kuriant šiuos modelius yra atsižvelgiama į DIS srityje keliamus saugumo, aplinkos, komunikacijos, energijos, naudojamų mazgų ir kt. reikalavimus. Sukurti modeliai aprašo galimus funkcinį ir nefunkcinį reikalavimų įgyvendinimo būdus (produktų šeimas), kurie gali būti taikomi kuriamai DIS.

Atsižvelgdami į DIS sudėtingumą, reikalavimų kompleksiškumą ir kūrimo iššūkius, mes pasiūlėme DIS aprašyti keletu bendrinių modelių, atspindinčių tam tikras reikalavimų grupes. Toliau, siekdami supaprastinti aprašymus ir modeliavimą, mes išskyrėme dvi reikalavimų grupes – funkcinis ir nefunkcinis reikalavimus. Tačiau mūsų sukurtą metodą galima naudoti bet kokiam reikalavimų grupių skaičiui. DIS produktų šeimas ir pakartotino panaudojimo galimybes aprašėme naudodami požymių modelius [Kang et al., 1998].

Taip pat mes manome, kad bendriniame nefunkcinį reikalavimų modelyje pateikti požymiai – saugumo lygis, energijos vartojimo ir aplinka – turi likti tokie, kaip pateikta 3.3a pav., nes jų pakanka perteikti nagrinėjamos DIS srities reikalavimus. Bendriniame funkcinį reikalavimų modelyje pateikti mazgus aprašantys požymiai turi turėti tokią



(a) Nefunkcinių reikalavimų požymių modelis



(b) Funkcinių reikalavimų požymių modelis

3.3 pav.: Pasiūlyti požymių modeliai

struktūrą, kaip pateikta 3.3b pav., nes mazgas yra DIS kūrimo vienetą, kuris yra sudarytas iš komunikacijos prietaiso, veiksmus atliekančio

prietaiso (pvz., jutiklio) ir tik šiam mazgui būdingos komunikacijos tipo – vienkryptės arba dvikryptės.

Paskutinis žingsnis šiame sluoksnyje yra sukurtų modelių verifikavimas – patikrinimas, ar jie yra teisingi, t. y. ar neturi nepasiekiamų požymių (patikrinimas atliekamas automatiniais įrankiais). Sukurti funkcinių ir nefunkcinių reikalavimų modeliai aprašo galimas nagrinėjamos DIS srities konfigūracijas (DIS variantiškumą), kurios gali būti naudojamas įgyvendinant konkrečią DIS.

3.3. Agregavimo fazė

Agregavimo fazėje modelių kūrimo fazėje sukurti funkcinių ir nefunkcinių reikalavimų požymių modeliai sujungiami – sudaromas vienas modelis, kuris aprašo nagrinėjamą DIS sritį. Agreguotas modelis aprašo visas galimas konfigūracijas (produktų šeimas) DIS kurti.

Agregavimas požymių modelius ir jų ribojimus traktuoja kaip atskirus komponentus, kuriuos reikia sujungti. Po agregavimo gauname naują modelį, kuris turi naują šakninį požymį (kurį siūlome pavadinti nagrinėjamos srities vardu) ir apima visus funkcinių ir nefunkcinių reikalavimų modelių požymius bei šių modelių ribojimus tarp požymių. Naujai sukurtas modelis leidžia pridėti naujus ribojimus tarp funkcinių ir nefunkcinių reikalavimų požymių. Pridėjus naujus ribojimus, reikia užtikrinti, kad jie neprieštarauja vieni kitiems, todėl reikia atlikti modelio verifikavimą. Jeigu modelis verifikuojamas sėkmingai, turime agreguotą modelį, kuris parodo visas galimas nagrinėjamos DIS srities kūrimo konfigūracijas. Konfigūracijų skaičius priklauso nuo nagrinėjamos DIS srities, nuo modelių detalumo ir ribojimų tarp požymių.

Agreguotas modelis aprašo visas srities galimas konfigūracijas (produktų šeimas), kurios gali būti naudojamos kuriant DIS, todėl toks modelis gali būti pakartotinai panaudotas kuriant kitas DIS toje pačioje srityje. Agreguotas modelis aprašo nemažai konfigūracijų, kurios nėra aktualios kuriamai sistemai, tad reikia palikti tik tas konfigūracijas, kurios yra aktualios, tai yra atliekama specializavimo fazėje.

3.4. Specializavimo fazė

Specializavimo fazėje sudaroma specializuota konfigūracija, iš kurios yra generuojamas kuriamos DIS karkasas. Specializuota konfigūracija gaunama agreguotą modelį sujungiant su kuriamos DIS specifikacijomis. Šioje fazėje agreguoto modelio požymiai, kurių nereikia kuriamai sistemai aprašyti, yra panaikinami. Taip pat dėl sistemos specifikacijų atsiranda naujų ribojimų tarp požymių. Sujungę agreguotą modelį ir DIS specifikacijas bei pridėję reikiamus ribojimus, gauname specializuotą

modelį, kurį reikia verifikuoti. Po verifikavimo turime modelį, kuris aprašo visas konfigūracijas (produktų šeimas), kurios tenkina kuriamos DIS specifikacijas ir kurios gali būti naudojamos DIS kurti. Kitaip sakant, šitas žingsnis yra skirtas susiaurinti agreguoto modelio aprašomą galimą DIS kūrimo variantiškumą (visos DIS srities) iki tenkinančio konkrečios DIS specifikacijos.

Specializuotas modelis aprašo kelias galimas konfigūracijas, kurios gali būti naudojamos DIS kurti. Kiekviena konfigūracija aprašo skirtingus aparatūrinius modulius, aparatūrinių modulių darbo režimus, komunikacijos protokolus, komunikacijos protokolų užtikrinimą saugumą ir t. t. Šios konfigūracijos – tai galimų DIS kūrimo konfigūracijų erdvė, kur kiekviena konfigūracija atitinka skirtingus DIS paslaugų kokybės reikalavimus. Todėl, turint tokią erdvę, reikia pasirinkti vieną konfigūraciją, kuri geriausiai atitinka kuriamos DIS paslaugų kokybės reikalavimus. Šiam tikslui mes siūlome naudoti Pareto optimizavimo metodą, kuris įvertina kiekvieną konfigūraciją pagal kuriamos DIS paslaugų kokybės reikalavimus ir pasiūlo geriausią konfigūraciją (specializuotą konfigūraciją) DIS kurti.

Parentant tinkamą specializuotą konfigūraciją yra naudojama komponentų bazė, kurioje yra duomenys apie aparatūrinę ir programinę įrangą (komunikacijos moduliai, sensoriai ir kita), kuri gali būti naudojama kuriant DIS. Kiekvienas komponentas yra aprašomas jam būdingomis charakteristikomis, pvz.: tikslumas, darbo atstumas, greitaveika ir t. t., kurios yra svarbios DIS ir gali turėti įtakos parentant specializuotą konfigūraciją. Iš galimos DIS kūrimo konfigūracijų erdvės parinkta specializuota konfigūracija aprašo tikslų komponentų sąrašą ir jų darbo režimus, kurie yra parentami iš komponentų bazės ir kurie turi būti naudojami kuriant DIS, siekiant užtikrinti reikiamą paslaugų kokybę.

3.5. Generavimo fazė

DIS kūrimo generavimo fazėje sugeneruojamas programinio kodo karkasas iš specializuotos konfigūracijos, gautos specializavimo fazėje. Programinio kodo generatorius programinį kodą generuoja iš specializuotos konfigūracijos, naudodant programinio kodo bibliotekas. Programinio kodo bibliotekose yra programinio kodo metodai, kur metodų grupė turi žymę, kokiam aparatūriniam komponentui jie tinka, taip pat bibliotekoje aprašomi metodai, skirti duomenims siūsti.

Programinio kodo generatorius kaip įvesties informaciją naudoja specializuotą konfigūraciją, analizuoja ją ir pagal joje rastą informaciją ieško atitinkamų komponentų. Pirmiausia yra nustatoma, kuri programinio kodo biblioteka turi būti naudojama. Kitame žingsnyje bibliote-

koje ieškoma programinių metodų, kurie aprašo specializuotoje konfigūracijoje pateiktų aparatūrinių komponentų veikimą.

Baigus metodų atrinkimą, yra gaunamas metodų sąrašas, kur kiekvienas atrinktas metodas yra priskirtas atitinkamam mazgui. Kitame kodo generavimo žingsnyje yra nustatomas formatas, koku atrinktas programinis kodas turi būti pateiktas, tai priklauso nuo programinės kalbos charakteristikų, ar bus naudojamos klasės, ar ne. Ši informacija yra paimama iš programinio kodo bibliotekos. Jeigu reikia naudoti klases, atrinktas programinis kodas yra suskaidomas į klases kaip atskirus failus, kur klasių vardai tokie, kaip mazgų, pateiktų specializuotoje konfigūracijoje, kitu atveju į atskirus failus, kur failų vardai tokie, kaip mazgų. Šie sugeneruoti failai yra kuriamos DIS karkasas.

3.6. Įgyvendinimo fazė

Šioje fazėje baigiama kurti DIS, kurios karkasas buvo sugeneruotas generavimo fazėje, praplečiant jį reikiamu funkcionalumu. Mūsų manymu, programinės įrangos kūrimo valdymo baigiamasis procesas [Li and Tang, 2014] geriausiai atitinka veiksmus, kurie yra atliekami šioje DIS kūrimo fazėje. DIS įgyvendinimo fazėje išskirti trys žingsniai: DIS karkaso testavimas, DIS įgyvendinimas ir DIS testavimas. Karkaso testavimo metu patikrinama, koks DIS funkcionalumas nebuvo sugeneruotas generavimo fazėje. Šis žingsnis yra sugeneruoto karkaso verifikavimas. Įgyvendinant DIS baigiama kurti iki galo veikianti sistema, papildant sugeneruotą karkasą reikiamais metodais ir loginiais ryšiais tarp metodų ir karkaso dalių. DIS testavimo žingsnyje yra testuojama sukurta DIS, ar ji veikia tinkamai.

Įvykdžius visas DIS kūrimo fazes, yra gaunama veikianti DIS.

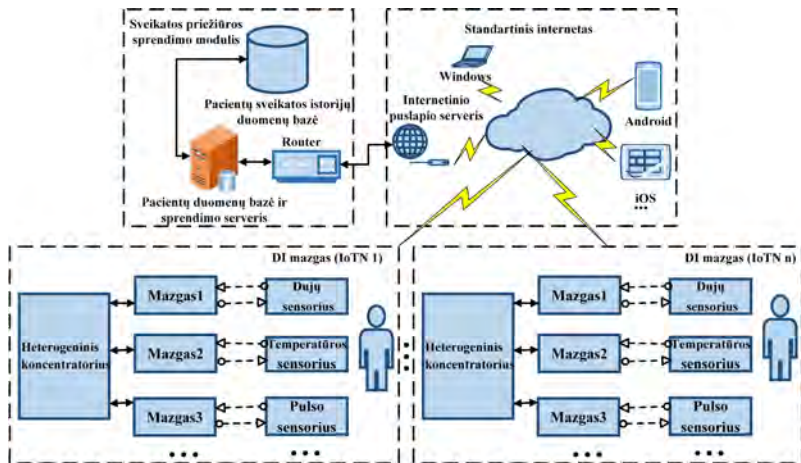
4. PROTOTIPO KŪRIMAS

Šiame skyriuje yra aprašomas pasiūlyto DIS kūrimo metodo panaudojimas kuriant DIS prototipą. Kaip prototipas buvo pasirinkta žmogaus sveikatos parametų stebėjimų sistema (ŽSPSS).

Kuriamos DIS principinė schema yra pavaizduota 4.1 pav. Kadangi mūsų pasiūlytas metodas yra skirtas įgyvendinti matavimų ir komunikacijos modulius, kuriant prototipą daugiausia dėmesio buvo skiriama žmogaus sveikatos parametų stebėjimo sprendimų srityje.

4.1. ŽSPSS modelių kūrimas

Žmogaus sveikatos parametų stebėjimų sistemų srities analizei atlikti buvo naudojami pasiūlyti bendriniai funkcinių ir nefunkcinių reika-



4.1 pav.: Žmogaus sveikatos parametrų stebėjimų sistemos principinė schema

lavimų modeliai (žr. 3.3 pav.). Analizuojant pasirinktą DIS sritį ir naudojant pasiūlytus požymių modelius, didžiausias dėmesys buvo skiriamas: mazgams, aparatūros platformoms ir komunikacijos protokolams, kurie yra arba gali būti naudojami šioje DIS srityje. Taip pat, naudodami pasiūlytus bendrinius reikalavimų požymių modelius, juos papildėme tik žemiausio lygio požymiais. Sukurti modeliai yra pateikti 4.2 pav., 4.2a pav. pavaizduotas sukurtas nefunkcinių reikalavimų požymių modelis nagrinėjamai žmogaus sveikatos parametrų stebėjimų sričiai. Kuriant nefunkcinių reikalavimų modelį, buvo pasirinktos penkios aparatūros platformos kaip galimos, kurios gali būti naudojamos DIS kurti. Kadangi nagrinėjamos DIS dažniausiai yra koncentruotos nedidelėje erdvėje, mazgų komunikacijos atstumai buvo pasirinkti nuo 0,2 m iki 10 m (žr. 4.2a pav.). 1 lentelėje yra pateiktos sukurto modelio charakteristikos, iš kurių matome, kad sukurtas modelis aprašo 562 185 galimas konfigūracijas (produktų šeimą), kurios gali būti naudojamos kuriamos DIS nefunkciniams reikalavimams įgyvendinti.

4.2b pav. yra pateiktas sukurtas funkcinių reikalavimų modelis. Nagrinėjamos DIS srities analizė parodė, kad šioje srityje naudojami mazgai dažniausiai naudoja belaidžius komunikacijos protokolus duomenims perduoti, todėl, kurdami funkcinių reikalavimų modelį, pasirinkome populiariausius belaidžius protokolus, kurie yra naudojami šiandien.

1 lentelė: Funkcinių, nefuncinių reikalavimų ir agreguoto modelių charakteristikos, gautos naudojant S.P.L.O.T įrankį

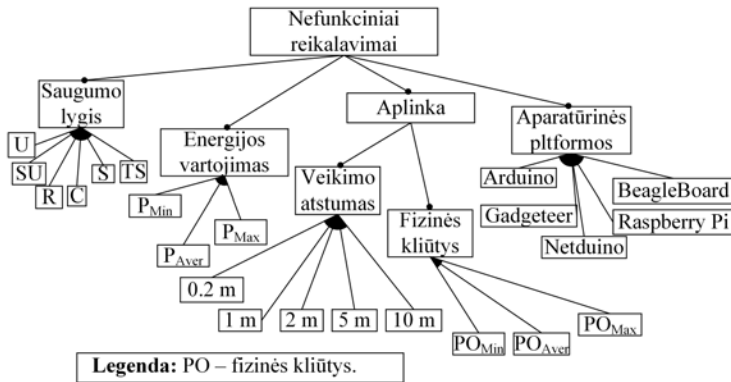
Nr.	Data	Nefuncinių reik. modelis	Funcinių reik. modelis	Agreguotas modelis
<i>Statistika</i>				
1	Požymių sk.	29	54	83
3	Ribojimai	9	10	27
<i>Rezultatai</i>				
1	Vientisumas	Vientisas	Vientisas	Vientisas
2	Mirę požymiai	0	0	0
<i>Metrikos</i>				
1	Konfigūracijų sk.	562185	1568352	4.34E7

Toliau analizuojant probleminę sritį, buvo nustatyta, kad joje galima / yra naudojami įvairiausi mazgai, todėl mes pasirinkome 6 populiariausius (žr. 4.2b pav.). Požymis *Duomenų siuntimo dažnis* aprašo, kaip dažnai mazgas siunčia duomenis. Mes pasirinkome 5 laikus, kurie, mūsų manymu, yra aktualiausi nagrinėjamai sričiai. Kiti požymiai buvo užpildyti atsižvelgiant analizės metu surinktą informaciją (žr. 4.2b pav.). Sukurto funkcinių reikalavimų modelio charakteristikos yra pateiktos 1 lentelėje, iš jos matyti, kad sukurtas modelis aprašo 1 568 352 galimas konfigūracijas (produktų šeimą), kurios gali būti naudojamos kuriamos DIS funkciniams reikalavimams įgyvendinti.

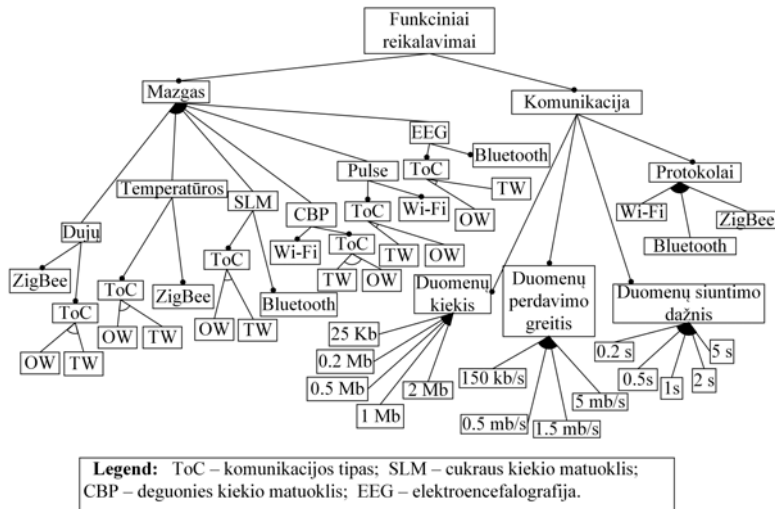
4.2. ŽSPSS modelių agregavimas

Kaip buvo minėta anksčiau, agregavimo fazė yra skirta agreguoti sukurtus funkcinių ir nefuncinių reikalavimų modelius (žr.4.2 pav.). Sukurtas naujas agreguotas žmogaus sveikatos parametrų stebėjimų srities modelis yra pavaizduotas 4.3 pav. Kaip matyti, agreguotas modelis sujungia visus funkcinių ir nefuncinių reikalavimų modelių požymius į vieną. Toks modelis visapusiškai aprašo žmogaus sveikatos parametrų stebėjimų sistemų sritį.

1 lentelėje yra pateiktos agreguoto modelio charakteristikos, iš kurių matyti, kad šis modelis sujungia ne tik agreguojamų modelių požymius, bet ir šių modelių ribojimus. Taip pat matome, kad atsirado papildomų ribojimų. Naujų ribojimų atsiranda dėl to, kad agreguotas modelis leidžia pridėti naujus ribojimus tarp funkcinių ir nefuncinių reikalavimų požymių (dėl modelio skaitomumo parodyti ne visi ribojimai). Agregavus ir pridėjus naujų ribojimų, modelis buvo verifikuojamas, nau-



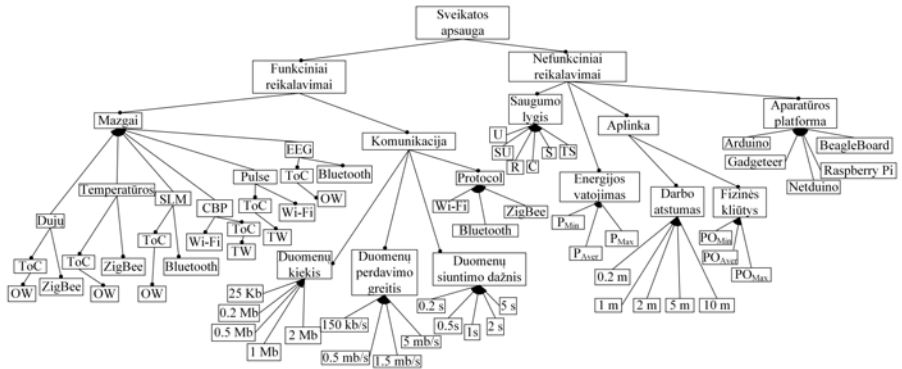
(a) Žmogaus sveikatos stebėjimo srities nefunkcinių reikalavimų modelis



(b) Žmogaus sveikatos stebėjimo srities funkcinių reikalavimų modelis

4.2 pav.: Sukurti nefunkcinių ir funkcinių reikalavimų modeliai

dojant S.P.L.O.T. įrankį. Agreguotas modelis aprašo $4.34E7$ konfigūracijas (produktų šeimą), kurios gali būti naudojamos ŽSPSS įgyvendinti. Atsižvelgiant į galimų konfigūracijų skaičių, galima teigti, kad agreguotas modelis gali būti panaudotas panašioms sistemoms kurti toje pačioje DIS srityje. Ne visos konfigūracijos, aprašytos šiame modelyje, yra tinkamos kuriamai DIS, norit susiaurinti konfigūracijų skaičių iki tinkamų



4.3 pav.: Žmogaus sveikatos stebėjimo srities agreguotas modelis

2 lentelė: Specializuoto modelio charakteristikos

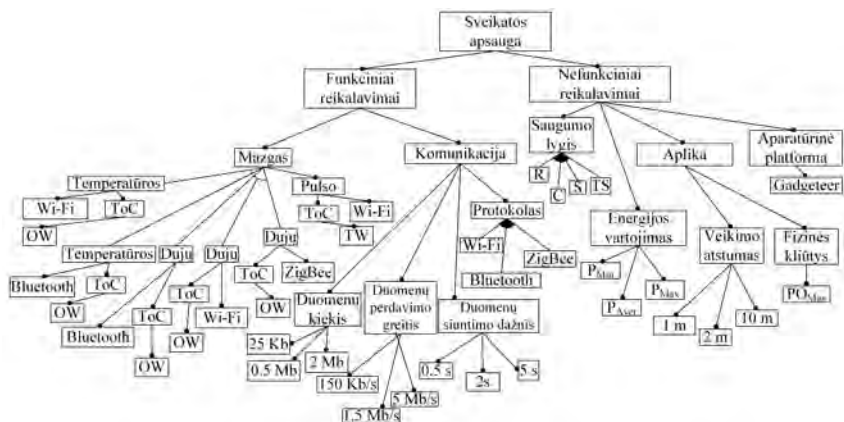
Nr.	Duomenys	Specializuotas modelis
<i>Statistika</i>		
1	Požymių sk.	64
2	Ribojimai	27
<i>Rezultatai</i>		
1	Vientisumas	Vientisas
2	Mirę požymiai	0
<i>Metrikos</i>		
1	Konfigūracijų sk.	120

kuriamam sprendimui, yra naudojama specializacija.

4.3. ŽSPSS modelio specializavimas

Modelio specializavimo fazėje agreguotas modelis specializuojamas pagal konkrečios ŽSPSS specifikacijas. Kuriamas DIS prototipas turi atitikti tokias specifikacijas: naudojami temperatūros, dujų ir pulso jutikliai, aparatūrinė platforma – Gadgeteer; komunikacijai gali būti naudojama Wi-Fi, Bluetooth ir Zig-Bee protokolai; saugumo lygis ne mažiau nei R; jutiklių veikimo nuotolis: pulso iki 10 m, dujų iki 5 m ir temperatūros iki 1 m; jutiklių sugeneruojamas duomenų kiekis: pulso iki 2 MB, dujų iki 25 KB, temperatūros iki 1,5 MB; duomenų siuntimo dažnis: pulso – 0,5 s, dujų – 2 s, temperatūros – 5 s; komunikacijos tipas: pulso – dvikryptis, dujų ir temperatūros – vienkryptis.

Įvertinus ŽSPSS keliamus reikalavimus ir tai, kad reikalavimuose



4.4 pav.: Kuriamo DI sprendimo specializuotas modelis

nebuvo pasakyta, koks komunikacijos protokolas turi būti naudojamas ryšiu su kuriuo mazgu, po specializavimo buvo gautas modelis, parodytas 4.3 pav. Šis modelis aprašo visas konfigūracijas, kurios tenkina kuriamos ŽSPSS reikalavimus ir kurios gali būti naudojamos kuriant šią DIS. Kaip matyti iš 2 lentelės, specializuotas modelis aprašo 120 galimų konfigūracijų. Šios konfigūracijos yra galima ŽSPSS kūrimo erdvė.

Kitame specializavimo žingsnyje iš gautos ŽSPSS kūrimo erdvės reikia parinkti vieną konfigūraciją, kuri geriausiai tenkina sistemos paslaugų kokybės reikalavimus. Pagrindiniai kriterijai, kuriais remiantis buvo vertinamos konfigūracijos, pateiktos ŽSPSS kūrimo erdvėje, pasirinkti: saugumas, energija ir duomenų siuntimo sparta. Konfigūracijoms vertinti buvo naudojamas Pareto metodas, aprašytas 3.4 skyriuje. Šiame žingsnyje požymių modelio buvo naudojamas S.P.L.O.T. įrankis ir sukurta DIS komponentų bazė. Gauta specializuota konfigūracija, kuri geriausiai atitinka kuriamos ŽSPSS paslaugų kokybės reikalavimus, pateikta 4.5 pav.

4.4. ŽSPSS karkaso generavimas

Generavimo fazėje buvo sugeneruotas ŽSPSS programinio kodo karkasas iš specializuotos konfigūracijos (žr. 4.5 pav.). Programinis kodo karkasas buvo generuojamas pagal 3.5 skyriuje pateiktas taisykles. Programinio kodo generavimo procesas yra pateiktas 4.6 pav.

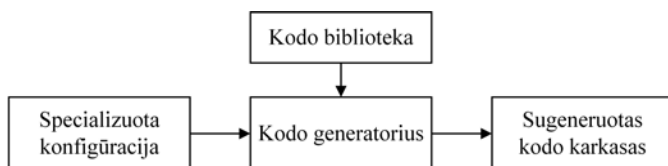
Kadangi generatorius naudoja programinio kodo bibliotekas, eksperimento tikslais buvo sukurta programinio kodo biblioteka, kurioje

```

<Healthcare>
  <Hardware platform> Gadgeteer</Hardware platform>
  <Pulse>
    <Action_module>Pulse_oximeter</Action_module>
    <Communication module_name = 'Wi-Fi_RS21' >
      <Security_level>WEP_128</Security_level>
    </Communication>
    <TypeOfCommunication>IW</TypeOfCommunication>
  </Pulse>
  <Temperature>
    <Communication module_name = 'Bluetooth' >
      <Security_level>Mode3</Security_level>
    </Communication>
    <TypeOfCommunication>OW</TypeOfCommunication>
  </Temperature>
  <Gas>
    <Communication module_name = 'ZigBee_XBee' >
      <Security_level>ENC-MIC-64</Security_level>
    </Communication>
    <TypeOfCommunication>OW</TypeOfCommunication>
  </Gas>
</Healthcare>

```

4.5 pav.: Specializuota konfigūracija



4.6 pav.: DIS karkaso generavimas

buvo surašyti programiniai metodai, skirti Gadgeteer aparatūrinei platformai. Generuojant programinio kodo karkasą, kaip tarpiniai rezultatai buvo gauti trys skirtingi metodų sąrašai, skirti skirtingiems mazgams (temperatūrai, dujoms ir pulsui, 4.1 pav.). Paskutiniame kodo generavimo žingsnyje atrinkti metodai buvo išskaidyti į tris klases kaip atskirus failus, kadangi Gadgeteer aparatūrinė platforma naudoja C# programavimo kalbą, kuri yra objektinė ir reikalauja klasių. Sugeneruoti programinio kodo failai (ŽSPSS karkasas) dalinai aprašo kuriamo žmogaus sveikatos parametrų stebėjimų sistemos veikimą.

4.5. Įgyvendinimo sluoksnis

Šioje DIS kūrimo fazėje sugeneruotas ŽSPSS karkasas buvo papildytas reikiamu kodu, taip užbaigiant sistemos įgyvendinimą. Kaip matyti iš specializuotos konfigūracijos (žr. 4.5 pav.), kuriama ŽSPSS turi naudoti Gadgeteer aparatūrinę platformą. Dėl šios priežasties kuriamam sprendimui įgyvendinti buvo pasirinkta naudoti Microsoft Visual Studio įrankius. Šie įrankiai buvo pasirinkti, nes palaiko C# programavimo kalbą ir suteikia galimybes programuoti Gadgeteer aparatūrinės platformos komponentus. Konkretūs aparatūriniai komponentai, kurie turi būti naudojami ŽSPSS kurti, yra pateikiami specializuotoje konfigūracijoje (žr. 4.5 pav.).

Sukurtos ŽSPSS eksperimentiniam tyrimui atlikti ir prototipui demonstruoti buvo sukurta internetinė paslauga, kuri surinko ir atvaizdavo siunčiamos žmogaus sveikatos parametrus.

5. IŠVADOS

1. Sukurtas DIS kūrimo metodas, pagrįstas požymių modeliais, leidžia aprašyti DIS produktų šeimas tam tikrose DI srityse.
2. Pasiūlyti bendriniai reikalavimų modeliai leidžia aprašyti DIS variantiškumą ir sudėtingumą, įvertinant technologijų heterogeniškumą ir kompleksinius paslaugų kokybės reikalavimus: saugumo, energijos sąnaudų, aplinkos veiksmus.
3. Specializuojant agreguotą modelį, gaunamas konkrečios DIS reikalavimus tenkinantis specializuotas požymių modelis. Specializuotas modelis aprašo DIS konfigūracijų erdvę, kuri tenkina kuriamos DIS specifikacijas.
4. DIS konfigūracijų erdvė tiriama ir įvertinama Pareto metodu. Gautas Pareto optimalus modelis geriausiai tenkina DIS keliamus reikalavimus ir yra naudojamas DIS karkasui generuoti.
5. Pasiūlytas metodas nuo analizuotų metodų skiriasi tuo, kad anksčiau sistemos kūrimo etape aprašo DIS produktų šeimą, pateikia konfigūracijų erdvę, kuri tenkina kuriamos DIS reikalavimus, ir įvertina šias konfigūracijas pagal DIS paslaugų kokybės reikalavimus, pateikdamas specializuotą konfigūraciją, kuri geriausiai atitinka reikalavimus.
6. Požymių modeliai, naudoti pasiūlytame metode, yra nepriklausomi

nuo taikymų srities. Todėl sukurti modeliai gali būti panaudoti kuriant įvairias DIS.

7. Pasiūlyti bendriniai funkcinių ir nefunkcinių reikalavimų požymių modeliai buvo naudoti kuriant DIS prototipą. Eksperimentai parodė, kad, naudojant šiuo modelius, DIS kompleksškumas ir galimų kūrimo konfigūracijų erdvė gali būti tinkamai aprašyti.

Literatūra

- [Anon et al., 2014] Anon, F., Navarathinarasah, V., Hoang, M., and Lung, C. H. (2014). Building a Framework for Internet of Things and Cloud Computing. In *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, pages 132–139.
- [Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey . *Computer Networks*, 54(15):2787 – 2805.
- [Ayala et al., 2015] Ayala, I., Amor, M., Fuentes, L., and Troya, J. M. (2015). A Software Product Line Process to Develop Agents for the IoT. *Sensors*, 15(7):15640–15660.
- [Berger et al., 2013] Berger, T., Rublack, R., Nair, D., Atlee, J. M., Becker, M., Czarnecki, K., and Wasowski, A. (2013). A Survey of Variability Modeling in Industrial Practice. In *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems*, VaMoS '13, pages 7:1–7:8, New York, NY, USA. ACM.
- [Chen et al., 2014] Chen, S., Xu, H., Liu, D., Hu, B., and Wang, H. (2014). A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective. *IEEE Internet Things J.*, 1(4):349–359.
- [Conejar and Kim, 2016] Conejar, R. J. and Kim, H.-K. (2016). Conceptual Framework for Mobile Device Product Line Security based on Internet of Thing. *International Journal of Hybrid Information Technology*, 9(7):411–418.
- [García et al., 2014] García, C. G., Espada, J. P., Valdez, E. R. N., and Díaz, V. G. (2014). Midgar: Domain-Specific Language to Generate Smart Objects for an Internet of Things Platform. *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 352–357.
- [Grace et al., 2016] Grace, P., Pickering, B., and Surridge, M. (2016). Model-driven interoperability: engineering heterogeneous IoT systems. *Annals of Telecommunications*, 71:141–150.

- [Gubbi et al., 2013] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Gener. Comput. Syst.*, 29(7):1645–1660.
- [IERC, 2009] IERC (2009). Internet of Things Strategic Research Roadmap. Technical report, European Research Cluster on the Internet of Things.
- [Kang et al., 1998] Kang, K. C., Kim, S., Lee, J., Kim, K., Shin, E., and Huh, M. (1998). Form: A feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, 5:143–168.
- [Kawamoto, 2016] Kawamoto, J. (2016). An implementation of privacy preserving stream integration system. In *2016 International Conference on Information Networking (ICOIN)*, pages 57–62.
- [Labioud et al., 2007] Labiod, H., Hossam, A., and Santis, C. D. (2007). *Wi-Fi, Bluetooth, Zigbee and WiMax*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Li et al., 2014] Li, L., Rong, M., and Zhang, G. (2014). An Internet of things QoS estimate approach based on multi-dimension QoS. In *Computer Science Education (ICCSE), 2014 9th International Conference on*, pages 998–1002.
- [Li and Tang, 2014] Li, L. and Tang, S. (2014). Some Reform Ideas for the Software Project Management Course. *Proceedings of the 3rd International Conference on Science and Social Research*.
- [Lopez et al., 2014] Lopez, L., Ozdemir, O., Kuemper, D., Stanca-Kaposta, B., Chainho, P., De, S., and Sasu, L.-M. (2014). Internet of Things Environment for Service Creation and Testing. Technical report, European Commission.
- [Patel and Cassou, 2015] Patel, P. and Cassou, D. (2015). Enabling high-level application development for the Internet of Things. *Journal of Systems and Software*, 103:62–84.
- [Patel et al., 2011] Patel, P., Pathak, A., Teixeira, T., and Issarny, V. (2011). Towards application development for the Internet of Things. In *ACM/IFIP/USENIX 12th International Middleware Conference*, page 5, Lisboa, Portugal.

- [Riedel et al., 2010] Riedel, T., Yordanov, D., Fantana, N., Scholz, M., and Decker, C. (2010). A model driven Internet of Things. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 265–268.
- [Saha et al., 2017] Saha, H. N., Mandal, A., and Sinha, A. (2017). Recent trends in the Internet of Things. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–4.
- [Salihbegovic et al., 2015] Salihbegovic, A., Eterovic, T., Kaljic, E., and Ribic, S. (2015). Design of a domain specific language and IDE for internet of things applications. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*, pages 996–1001.
- [Shaoshuai et al., 2011] Shaoshuai, F., Wenxiao, S., Nan, W., and Yan, L. (2011). MODM-Based Evaluation Model of Service Quality in the Internet of Things. *Procedia Environmental Sciences*, 11:63–69.
- [Smith, 2012] Smith, I. (2012). The Internet of Things 2012: New Horizons. Technical report, European Research Cluster on the internet of thongs.
- [Vermesan and Friess, 2014] Vermesan, O. and Friess, P., editors (2014). *Internet of Things - From Research and Innovation to Market Deployment*. River Publishers.

AUTORIAUS PUBLIKACIJŲ SĄRAŠAS

STRAIPSNIAI

Tarptautinėse duomenų bazėse esančiuose mokslo leidiniuose paskelbti straipsniai

Mokslinės informacijos instituto duomenų bazės „ISI Web of Science“ leidiniuose, turinčiuose citavimo indeksą

1. Venčkauskas, Algimantas; Jusas, Nerijus; Kazanavičius, Egidijus; Štuitkys, Vytautas. Identification of dependency among energy consumption and Wi-Fi protocol security levels within the prototype module for the IoT // Elektronika ir elektrotechnika = Electronics and electrical engineering. Kaunas: KTU. ISSN 1392-1215. 2014, Vol. 20, no. 6, p. 132-135. [Science Citation Index Expanded (Web of Science); Inspec; Computers & Applied Sciences Complete; Central & Eastern European Academic Source]. [Indėlis: 0,250]. [IF (E): 0,561 (2014)]].
2. Venčkauskas, Algimantas; Jusas, Nerijus; Kazanavičius, Egidijus; Štuitkys, Vytautas. An energy efficient protocol for the internet of things // Journal of electrical engineering-Elektrotechnický časopis. Berlin: De Gruyter. ISSN 1335-3632. 2015, vol. 66, iss. 1, p. 47-52. [Science Citation Index Expanded (Web of Science); Scopus; Inspec]. [Indėlis: 0,250]. [IF (E): 0,407 (2015)]].
3. Venčkauskas, Algimantas; Štuitkys, Vytautas; Jusas, Nerijus; Burbaitė, Renata. Model-driven approach for body area network application development // Sensors. Basel: MDPI AG. ISSN 1424-8220. 2016, vol. 16, iss. 5, Article 670, p. [1-22]. [Science Citation Index Expanded (Web of Science); Academic Search Complete]. [Indėlis: 0,250]. [IF (E): 2,033 (2015)]].
4. Venčkauskas, Algimantas; Štuitkys, Vytautas; Toldinas, Jevgenijus; Jusas, Nerijus. A model-driven framework to develop personalized health monitoring // Symmetry. Basel: MDPI AG. ISSN 2073-8994. 2016, vol. 8, iss. 7, article 65, p. [1-18]. [Science Citation Index Expanded; Current Contents (Physical, Chemical & Earth Sciences); Academic Search Alumni Edition; Academic Search Complete; Academic Search Elite; Academic Search Premier; Academic Search Research & Development]. [Indėlis: 0,250]. [IF (E): 0,841 (2015)]].
5. Venčkauskas, Algimantas; Štuitkys, Vytautas; Damaševičius, Robertas; Jusas, Nerijus. Modelling of Internet of Things units for estimating security-energy-performance relationships for quality of service and environment awareness // Security and communication networks [elektroninis išteklius]. Hoboken, NJ: John Wiley & Sons. ISSN 1939-0114. 2016, vol. 9, iss. 16, p. 3324-3339. [Science Citation Index Expanded (Web of Science); Current Contents (Engineering, Computing & Technology)]. [Indėlis: 0,250]. [IF (E): 0,806 (2015)]].

Kitų tarptautinių duomenų bazių leidiniuose

1. Venčkauskas, Algimantas; Jusas, Nerijus; Toldinas, Eugenijus; Kazanavičius, Egidijus. Security level versus energy consumption in wireless protocols for internet of things // Information and software technologies : 20th international conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014 : proceedings / Giedre Dregvaitė; Robertas Damaševičius (Eds.) ; Kaunas University of Technology. Cham: Springer, 2014. (Communications in computer and information science, 465, ISSN 1865-0929), ISBN 9783319119571. p. 419-429. [Conference Proceedings Citation Index; SpringerLINK].

Kitos publikacijos

1. Jusas, Nerijus; Venčkauskas, Algimantas; Štuitkys, Vytautas. Model driven framework to develop the IoT-based healthcare applications // 12th annual international conference on information technology and computer science, 16- 19 May 2016, Athens, Greece : abstract book / edited by Gregory T. Papanikos. Athens: Athens Institute for Education and Research, ISBN 9789605980474. p. 24. [Sc. fields: 07T]. [0,333]].

Sąrašo sudarymo data: 2016-11-10

INFORMACIJA APIE AUTORIŲ

Išsilavinimas:

2006–2010 Įgijo informatikos inžinerijos bakalauro kvalifikacinį laipsnį Kauno technologijos universitete, Informatikos fakultete.

2010–2012 Įgijo informacijos saugos magistro kvalifikacinį laipsnį Kauno technologijos universitete, Informatikos fakultete.

2012–2016 Studijavo Kauno technologijos universiteto doktorantūros studijose (informatikos inžinerija OTT).

Darbo patirtis:

2012–2013 Vyresnysis inžinierius Kauno technologijos universiteto informatikos fakultete.

Nuo 2013 Kauno technologijos universiteto kompiuterių katedros lektorius.

Mokslinių interesų sritys: informacijos ir informacinių technologijų sauga, daiktų interneto technologijos, kompiuterių tinklų saugos valdymas.

E. paštas: nerijus.jusas@ktu.lt

RÉSUMÉ

Relevance of the work

Technology advances have resulted in our modern-day life and work in a digital world surrounded with the modern technology infrastructure – multiple devices integrated within networks along with computers, mobile devices, sensor networks, etc. which have become a commodity of our contemporary lives. In the nearest future, however, not only humans and computers but also everyday life items will be interconnected to create the new computing infrastructure – The Internet of Things (IoT). The goal of the IoT is to enable things to be connected "any-time, any-place, with anything and anyone ideally using any path/network and any service". Semantically, the IoT means a new highly heterogeneous worldwide network of interconnected objects uniquely addressable, based on standard communication protocols. This move from 'interconnected computers' to 'interconnected things' is a great challenge for the Information-Communication Technology (ICT) practitioners, scientists and the society in the whole. As a response to the challenge, an extremely wide stream of research is being provided worldwide.

In the case of its application, the IoT is to be considered as an information-communication technology with smart features and enhanced capabilities. From this viewpoint, the IoT stands for a huge infrastructure containing physical objects such as sensors, actuators, etc. that are self-identifiable to other devices; thus, being connected over the Internet, these objects enable communication and continuous transmission of the collected or control data over the nodes of a network via the Internet. Depending on the application, diverse sensors and other devices are served in collecting the data, on whose basis, the functionality of the application is built.

Typically, security/privacy, energy-awareness and environmental factors represent the major constraints in such applications. The first aspect is due to the possibility that the data can be launched and changed, e.g., during the transfer sessions. The second aspect is due to the use of battery-charged devices within the network. The third aspect deals with the noises influencing data transfer. The outlined factors are highly interrelated and extremely complex in their own way. For example, there are a variety of communication protocols designed to ensure different levels of security. The more complex a protocol is (in terms of the complexity of the encryption algorithms used), the more energy is required to ensure the required level of security. The same is true of environmental

noises. All these factors, when considered together, predefine the quality of service (QoS) of the application. Therefore, QoS should appear as a basic non-functional requirement in designing the IoT applications. QoS requirements for IoT applications are very different even when the applications are very similar.

The IoT application itself consists of internal nodes (sensors, actuators, and other devices). Typically, the number of nodes ranges from a few to a dozen nodes. This depends on the type of each IoT application (e.g., a smart house requires fewer sensors than a smart city). The nodes may be combined into groups in order to cover different aspects of the same application, or even different but related applications. Therefore, the functionality and structure of a node can differ significantly, depending on the requirements of a particular application. On the other hand, there might be identical nodes (e.g., for ensuring better performance, higher reliability, etc.).

As the complexity of systems is steadily increasing, application development approaches should rely on the successful ones already in use in the industry, such as Product Line Engineering (PLE), Model Driven Application Development (MDAD) or Model-Driven Product Lines (MDPL). The PLE approach is defined as a methodology intended to develop a family of related products in an efficient way, taking full advantage of the products' commonality and variability aspects. It is also concerned with the use of the variability management. MDAD raises the abstraction level of the typical application development by focusing on modelling and automated code generation from the models. Models of an application are specified by high-level feature modelling languages and using model transformations that typically are defined as model-to-program transformation which are converted into the application. Therefore, the model-driven application development approach is centered on the use of models and their transformations. MDPL combines the abstraction capability of MDAD and the variability management capability of PLE. MDPL uses models with the intention to present the possible variability in order to implement the application in the given domain where the variability is usually modelled by using feature models.

In IoT applications, due to their novelty and specificity, PLE approaches are not yet exploited in full to ensure better quality, higher productivity, more flexible adaptability and reuse despite the fact that IoT applications are domain-specific [Lopez et al., 2014] and share many common features in it. Therefore, the aim of the thesis is to analyze and disclose the potential of the product line engineering approach for designing IoT-oriented applications. Currently, the main prob-

lems in designing those applications are as follows: insufficient extent of automation, inadequate capabilities for process reuse, integration and adaptation. Furthermore, so far, designers have largely ignored the multiplicity and synergy of non-functional requirements (security and energy requirements, heterogeneity of devices and communication protocols) in designing their systems.

Object of the thesis

The object of this research is a feature modelling based method for the development of IoT-oriented applications product line.

Aim of the thesis

The aim of the thesis is the development of IoT-based applications in problematic domain(s) by using product line methodologies while accounting for IoT applications challenges: the complexity of requirements (security, energy consumption), the heterogeneity of technology, and the factor of the environment.

Tasks of the thesis

The main tasks of this thesis are:

1. To investigate the typical requirements and challenges of IoT applications development;
2. To propose an IoT applications product line development method based on feature modelling;
3. To perform transformations of feature models and specialized configuration to the framework of IoT application;
4. To produce practical implementation of the proposed method by implementing IoT application.

Scientific novelty

The thesis has achieved the following innovative results:

1. A novel feature model-based method for the development of IoT applications product lines has been proposed.
2. Generic functional and non-functional requirements feature models have been proposed which are used to present the variability of possible requirements of the IoT application in the application domain at an early stage of the application development.

3. The mapping procedure to map the problem domain variability presented by the feature model with the requirements of the specific IoT application has been proposed.
4. An approach to connect higher-level models with the generation level to produce IoT applications has been proposed.

Practical value

The creation of IoT-based applications is a very complex and time-consuming process because, during this process, software, hardware, the working modes of the hardware and the environmental factors must be combined in order to ensure the best performance (QoS) of an application in a given situation. Moreover, IoT applications are domain-specific and share many common features in the same domain [Chen et al., 2014], thus the product line technologies for the development of IoT applications can be used. PL methods allow to reuse the created application components for the development of another application. Considering that, IoT applications development methods with the following properties have been proposed:

- To present possible design space to implement IoT applications in a specific domain;
- To choose one implementation variant from the design space of some IoT application which meets the application's requirements optimally;
- To generate a framework of IoT application from the selected implementation variant.

The research results are included into project No. VP1-3.1-ŠMM-08-K-01-018 "Research and Development of Internet Technologies and Their Infrastructure for Smart Environments of Things and Services" funded by the EU SA.

Thesis statements

1. IoT applications development methods based on feature models can be used to develop the product line of IoT application(s) in the specific IoT domain.
2. The generic feature models present: the complexity of requirements (security, energy consumption), the heterogeneity of technology, and the environmental factors of the problematic IoT domain.

3. The result of generic feature models transformations is a Pareto optimal model of IoT applications which is used to generate a framework of the IoT application.

Scientific approval

All of the results presented in the thesis are original; they have been presented in 4 internationally referred "ISI Web of Science" scientific journal publications and 2 other publications in informatics, electronics and software engineering peer reviewed journals and proceedings.

The experimental results were presented and discussed in 3 international conferences:

1. 18th *International conference Electronics 2014*, Palanga;
2. 20th *International conference on information and software technologies*, ICIST 2014, Druskininkai;
3. 12th *Annual International Conference on Information Technology & Computer Science 2016*, Athens.

Thesis organization

The thesis consists of 5 chapters:

Chapter 1 is an introduction providing a short summary of the work's novelty, aim and objectives. This chapter includes a brief identification of the main problems in the IoT are and the motivation of the work.

Chapter 2 performs a thorough review of *Internet of Things*. This chapter describes in detail IoT applications, IoT applications development challenges, QoS of IoT applications and methods used for the development of IoT applications. The purpose of this chapter is to introduce the reader with the terminology and IoT applications characteristics and specifics which will be exploited and referred to in further chapters.

Chapter 3 presents a proposed feature model-based method intended to implement the IoT-based application. It starts with an overview of the proposed IoT application development method. The proposed method consists of five phases: development of models, aggregation, specialization, code generation and implementation. The processes of each phase are extensively described in a specific subsection.

Chapter 4 presents a prototype which implements the proposed IoT application development method presented in Chapter 3. It includes presentation of a case study of the implementation of IoT-based healthcare BAN application.

Chapter 5 is the concluding chapter where the proposed solution and contributions are summarized.

CONCLUSIONS

In this thesis, we have studied the model-driven method for Internet of things applications development, which is characterized by the opportunity to evaluate the following challenges of such applications: security and privacy; energy-awareness; environmental factors; diversity of sensors, actuators, communication protocols.

The relevant solution as the response to the emerging challenges is the use of prototyping combined with modern model-driven methodologies in designing the systems. Due to these facts, we have proposed a multi-layered IoT-based application development method which uses feature models.

1. The proposed IoT applications development method based on feature models is used to present possible configurations (different product lines) which can be used for the development of the IoT application in the specific IoT domain. Thus, the final application can be customized to meet the specifications of specific requirements.
2. The proposed generic feature models (functional and non-functional requirements) allow us to present: the variability of configurations (different PLs) to implement the application in a specific domain and the complexity of IoT-based applications with respect to security and energy requirements, environmental factors and the heterogeneity of devices and communication protocols.
3. Narrowing the variability space (PLs) of the aggregated feature model through the model specialization enables us to adapt configurations to the needs of the developing application requirements. This results in the creation of the specialized feature model, which presents the design space of configurations (PLs) meeting the specifications of IoT application which can be used for the implementation of the specific IoT-based application.
4. The result of design space exploration presented by the specialized feature model is a Pareto optimal feature model which presents the application which should be developed best and is used to generate a framework of the IoT application.

5. The proposed method is different from the analysed IoT applications development methods presenting the variability of configurations (PLs) which can be used for the implementation of the specific IoT application, evaluating each configuration according to the QoS requirements of application and providing a single best configuration to implement the given application.
6. Feature modelling and model transformations used in the proposed method, are independent from the IoT applications. Thus, the created aggregated feature models can be reused to implement other applications in the same domain or similar applications in other domains by skipping the modelling layer which decreases the application development time.
7. The proposed generic functional and non-functional requirements feature models were used in the case study for the development of the IoT-based healthcare BAN application. The experiment results showed that the complexity of the BAN layer and configurations (PLs) variability to implement the IoT-based healthcare application's BAN layer could be presented by using the proposed generic models. In addition, the Pareto optimal model for IoT-based healthcare BAN application framework generation could be created by using the proposed feature models.

UDK 004.738.5(043.3)

SL344. 2017-05-15, 2,5 leidyb. apsk. l. Tiražas 50 egz.

Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas