

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

**Naujos kartos sudėtingų informacinių sistemų
dekomponavimas**

Decomposition of next generation complex information systems

Magistro baigiamasis darbas

Atliko: Gediminas Krasauskas (parašas)

Darbo vadovas: dr. Audronė Lupeikienė (parašas)

Recenzentas: doc. dr. Vytautas Čyras (parašas)

Vilnius – 2021

Nuoširdžiai dėkoju savo darbo vadovei dr. Audronei Lupeikienei
už konstruktyvią kritiką, produktyvų reiklumą ir pagalbą.

Santrauka

Šiame darbe pateikiami sudėtingų informacinių sistemų dekomponavimo, būtino siekiant mažinti artefakto, kurį reikia įgyvendinti, sudėtingumą, tyrimo rezultatai. Programų sistemų kūrimo srityje yra atlikta įvairių tyrimų, skirtų dekompozicijos procesui analizuoti ir kiekybiniais dekomponavimo rezultatų vertinimams atlikti. Tačiau programų sistemų kūrime naudojami „skaidyk ir valdyk“ metodai negali būti tiesiogiai taikomi sudėtingoms informacinėms sistemoms dekomponuoti, pamatuoti dekompozicijos gerumą ir palyginti su kitomis alternatyvomis. Pasiūlytasis dekomponavimo kortų metodas, kuris apima dekompozicijos vertinimą atsižvelgiant į kiekybinius atributus, charakterizuojamas kaip hierarchinis ir nedarantis konkrečių techninių sprendimų aukštam abstrakcijos lygmenyje. Gauti rezultatai sudaro prielaidas tolesniems tyrimams vykdyti.

Raktiniai žodžiai: sudėtingos informacinės sistemos, dekomponavimo būdai, sisteminė dekomponavimo procedūra, dekomponavimo kortų metodas.

Summary

This paper presents the results of a study of the decomposition of complex information systems necessary to reduce the complexity of the artifact to be implemented. In the field of software systems development, various studies have been conducted to analyze the decomposition process and to perform quantitative evaluations of decomposition results. However, the „divide and conquer“ methods used in the development of application systems cannot be directly applied to decompose complex information systems, measure the appropriateness of decomposition and compare it with other alternatives. The proposed decomposition card method, which involves the evaluation of decomposition in terms of quantitative attributes, is characterized as hierarchical and does not make specific technical decisions at a high level of abstraction. The obtained results create preconditions for further research.

Keywords: complex information systems, ways of decomposition, systematic decomposition procedure, decomposition cards method.

TURINYS

IVADAS	7
1. NAUJOS KARTOS SUDĒTINGOS INFORMACINĒS SISTEMOS	10
1.1. Sudėtinga sistema	10
1.2. Naujos kartos informacinė sistema	11
1.3. Naujos kartos informacinės sistemos savybės	12
1.4. Naujos kartos informacinės sistemos abstrakti architektūra	13
2. SUDĒTINGŲ INFORMACINIŲ SISTEMŲ DEKOMPONAVIMAS	15
2.1. Dekomponavimo samprata ir svarba	15
2.2. Dekomponavimo etapai	16
2.3. Dekomponavimo procesas sistemų kūrimo gyvavimo cikle	17
2.4. Dekomponavimo būdų klasifikacija	19
2.5. Dekomponavimo būdai	22
2.5.1. Probleminiais freimais grindžiama dekompozicija	22
2.5.2. Funkcionalumu grindžiama dekompozicija	23
2.5.3. Agentais grindžiama dekompozicija	24
2.5.4. Dalykine sritimi grindžiama dekompozicija	25
2.5.5. Kintamumu grindžiama dekompozicija	26
2.5.6. Servisais grindžiama dekompozicija	27
2.5.7. Surinkimu grindžiama dekompozicija	27
2.6. Sistemos reikalavimų pavyzdinis katalogas	28
2.6.1. Funkcinės savybės	28
2.6.2. Direktyvinės savybės	29
2.6.3. Kokybinės savybės	30
2.7. Dekomponavimo kriterijai	31
3. DEKOMPONAVIMO KORTŲ METODAS	32
3.1. Originali kokybės funkcijų sklaidos metodika	32
3.2. Kortų pritaikymas dekomponavimo procesui	33
3.3. Informacinių sistemų dekomponavimo kortų metodas	36
3.3.1. Dekomponavimo kortos turinys	37
3.3.2. Kairioji dekomponavimo kortos dalis	38
3.3.2.1. Savybių sąrašas	39
3.3.2.2. Identifikatorius	39
3.3.2.3. Prioriteto reitingas	39
3.3.2.4. Santykinis prioriteto reitingas	40
3.3.2.5. Sąveikų matrica	41
3.3.3. Viršutinė dekomponavimo kortos dalis	42
3.3.3.1. Sudedamųjų dalių sąrašas	42
3.3.3.2. Identifikatorius	43
3.3.3.3. Ryšių matrica	43
3.3.4. Centrinė dekomponavimo kortos dalis	45
3.3.4.1. Atsakomybių matrica	46
3.3.5. Dešinioji dekomponavimo kortos dalis	49
3.3.5.1. Atsakingų dalių kiekis	49
3.3.5.2. Atsakingų dalių įverčių suma	50
3.3.5.3. Atsakingų dalių įverčių vidurkis	50
3.3.5.4. Alternatyvų savybių atributai	51
3.3.6. Apatinė dekomponavimo kortos dalis	51

3.3.6.1. Svarbos reitingas	52
3.3.6.2. Santykinė svarba	52
3.3.6.3. Atsakomybių kiekis	53
3.3.6.4. Atsakomybių verčių suma	54
3.3.6.5. Atsakomybių verčių vidurkis	54
3.3.6.6. Ryšių kiekis	55
3.3.6.7. Ryšių verčių suma	55
3.3.6.8. Ryšių verčių vidurkis	56
3.3.6.9. Sudedamųjų dalių kiekis	56
3.3.6.10. Svarbos reitingų suma	57
3.3.6.11. Svarbos reitingų mediana	57
3.3.6.12. Santykinės svarbos rangas	58
3.3.6.13. Atsakomybių kiekis	59
3.3.6.14. Atsakomybių verčių suma	59
3.3.6.15. Atsakomybių verčių vidurkio rangas	60
3.3.6.16. Ryšių kiekis	61
3.3.6.17. Ryšių verčių suma	61
3.3.6.18. Ryšių kiekio rangas	62
3.3.6.19. Alternatyvų elemento atributai	62
3.4. Dekomponavimo kortų metodo apibendrinančios pastabos	63
3.4.1. Dekomponavimo kortų metodo ribojimai	63
3.4.2. Dekomponavimo kortų metodo plėtojimo kryptys	65
4. DEKOMPONAVIMO KORTŲ METODO TAIKYMAS	67
4.1. Argumentų pagrįstumas	67
4.2. Metodo taikymo pavyzdys	69
4.2.1. Dekomponuojamos sistemos pavyzdys	69
4.2.2. Pirmoji dekompozicijos alternatyva	71
4.2.3. Antroji dekompozicijos alternatyva	73
4.2.4. Dekompozicijos alternatyvų palyginimas	76
REZULTATAI IR IŠVADOS	81
SAŲOKŲ APIBRĖŽIMAI	83
SANTRUMPOS	84
LITERATŪRA	85
PRIEDAI	91
1 priedas. Dekomponavimo kortų metodą realizuojančio įrankio metaduomenys	91

Įvadas

Tyrimo objektas – naujos kartos sudėtingų informacinių sistemų dekomponavimas (angl. *decomposition*).

Tyrimo naujumas ir aktualumas. Greiti technologiniai, ekonominiai ir socialiniai pokyčiai aplinkoje, kurioje funkcionuoja šiuolaikinės organizacijų informacinės sistemos, atitinkamai lemia spartesnę tų sistemų vystymąsi [NFT16; RV16]. Atsiranda reikalavimai apdoroti vis didesnius duomenų srautus ir informacijos kiekius [RV16]. Sistemoms, sudarytoms iš daugelio komponentų, tenka bendrauti su vis didesniu kiekiu išorinių sistemų, verslas kelia ambicingesnius reikalavimus, prisidedančius prie sistemų sudėtingumo [XL05]. Taigi, reikalaujama kurti sudėtingas sistemas, kurios nebegali būti projektuojamos taikant tradicinius sprendinius.

Dėl didelio tarpusavyje sąveikaujančių sistemos dalių kiekio, jų tarpusavio priklausomybių, ir kitų aspektų šiuolaikinių sistemų modeliavimas tampa itin sudėtingu procesu. Be to, dažnu atveju sistemų kokybė priklauso nuo pasirinktos architektūros ir inžinerinių sprendimų [SLL⁺18], o architektams reikalingas supratimas apie sistemas yra sunkiai įsivaizduojamas be dekomponavimo proceso [Bro01].

Tačiau dekomponavimo veikla nėra triviali. Skirtingi sistemų architektai tos pačios sistemos atžvilgiu priims nevieningus dekomponavimo sprendimus dėl skirtingų įgūdžių ir įsivaizdavimo apie potencialią sistemą [CHL⁺11]. Visa tai dėl to, kad nėra visuotinai pripažintų universalių gairių ir standartų, kaip dekomponuoti sudėtingas sistemas [CHL⁺11]. To pasekmė – dažnai netinkamai suprojektuotos sistemos: užsakovų poreikiai nėra įgyvendinami, o sistemų sudėtingumas tampa sunkiai valdomas. Šios priežastys gali nulemti informacinių sistemų kūrimo projektų nesėkmę [WD13]. Atsižvelgus į visą tai galima tvirtinti, kad organizacijų naujos kartos sudėtingų informacinių sistemų dekomponavimas yra aktuali problema.

Laikui bėgant informacinės sistemos (angl. *information systems*) įgaudavo įvairias formas pagal tuo laikmečiu vyravusias tendencijas, kurios dar vadinamos kartomis [PZJ⁺16; RV16]. Aštuntojo dešimtmečio informacinės sistemos būdavo sutelktos į programas, dominuodavo lokalių sistemų [PZJ⁺16]. Tuo tarpu XXI a. trečiame dešimtmetyje pagreitį įgauna naujos kartos sistemos.

Naujos kartos informacinės sistemos – tai kiber-fizinės proaktyvios sistemos, kurios sutelktos į visumą: sąveikos tinklus tarp vartotojų, duomenis, daiktus ir paslaugas, saugų atvirumą ir gebėjimą bendrauti su išorinėmis sistemomis ir esybėmis [PZJ⁺16]. Tokių sistemų dekompozicijos nėra plačiai išnagrinėtos – nėra aišku, pagal kokius kriterijus ir kokiais būdais tokias sistemas dekomponuoti yra tinkamiausia. Prasmingi rezultatai šioje srityje galėtų palengvinti sistemų architektų ir

projektuotojų darbą ir užtikrinti kuriamų sistemų kokybę.

Vienas iš galimų būdų išspręsti dekomponavimo metu kylančias problemas – sukonstruoti pavyzdinę sistemine dekomponavimo procedūrą, kurios rezultatų sėkmingumo nenulemtų mažesnė architektų kvalifikacija bei patirtis. Tokia procedūra turėtų sudaryti galimybes sukurti sistemos architektūrinius sprendinius, kurie nepriklausytų nuo konkrečių architektūros stilių bei techninių ribojimų. Taip pat architektui turėtų būti sudarytos galimybės užtikrinti sistemos reikalavimus, įvertinti dekompozicijos sudėtingumą ir analizuoti duomenis sprendimų priėmimo metu.

Šio darbo **tikslas** – ištirti sudėtingų sistemų dekomponavimą ir tuo remiantis sudaryti pavyzdinę (angl. *reference*) sistemine procedūrą sudėtingoms informacinėms sistemoms dekomponuoti, apimančią dekompozicijų vertinimą ir lyginimą.

Siekiant įgyvendinti užsibrėžtą tikslą, buvo išskelti šie **darbo uždaviniai**:

1. Išanalizuoti dekomponavimo procesą ir galimus dekompozicijos būdus, taikomus kuriant sudėtingas informacines sistemas;
2. Sukonstruoti naujos kartos sudėtingų informacinių sistemų dekomponavimo metodą (kitaip tariant, pavyzdinę sistemine dekomponavimo procedūrą);
3. Atlikti sukurto metodo kokybinį vertinimą.

Tyrimas buvo vykdytas pagal tokią **metodiką**:

- *1 uždavinys*. Atliktas apžvelgiamasis tyrimas siekiant išanalizuoti panašią problemą nagrinėjusių tyrėjų gautus rezultatus.
- *2 uždavinys*. Vykdytas teorinis tyrimas, kurio metu pavyzdinei sisteminei procedūrai sudaryti buvo panaudotas tyrimo konstravimu metodas;
- *3 uždavinys*. Buvo vykdomas atvejo tyrimas (angl. case study research). Tai vienintelis galimas metodas gautam rezultatui įvertinti, nes dėl reikiamo šiam tyrimui darbo apimties ir turimų išteklių santykio negalima užtikrinti tinkamos imties ir tuo pačiu išsamaus rezultato patikrinimo

Darbo struktūra. Pirmame skyriuje apžvelgiamos literatūroje sutinkamos sudėtingų sistemų bei naujos kartos informacinių sistemų sampratos, apibrėžimai ir charakteristikos; antrame skyriuje tiriamas dekomponavimo procesas: prasmė, sudaromieji etapai, lokalizavimas sistemų gyvavimo cikle, taip pat išanalizuota viena iš galimų dekomponavimo būdų klasifikacijų bei atlikta konkrečių dekomponavimo būdų literatūros analizė, apimanti jų trūkumus ir privalumus; trečiajame skyriuje pateikta praktinė tyrimo dalis: sukonstruotas dekomponavimo kortų metodas sudė-

tingoms informacinėms sistemoms dekomponuoti, aprašytos kortos dalys bei pateikti nurodymai ir paaiškinimai metodo taikymui. Ketvirtame skyriuje pateikiamas teorinių argumentų pagrindimas ir dekomponavimo kortų metodo taikymas: konkrečios įsivaizduojamos sistemos pavyzdžio dekomponavimas, dviejų alternatyvių sistemos dekompozicijų vertinimai bei jų tarpusavio palyginimas.

Darbo aprobavimas. Straipsnis, apimantis dalį magistro baigiamojo darbo rezultatų buvo priimtas į „Lietuvos magistrantų informatikos ir IT tyrimai“ konferenciją, vykusią 2021 m. gegužės 14 d. Konferencijos metu autorius pristatė tyrimo rezultatus, kurie buvo publikuoti oficialiame leidinyje [Kra21].

1. Naujos kartos sudėtingos informacinės sistemos

Šiame skyriuje nagrinėjama sudėtingos sistemos samprata, naujos kartos informacinė sistema, jos savybės ir pavyzdinė abstrakti architektūra.

1.1. Sudėtinga sistema

Sudėtingų sistemų (angl. *complex systems*) sąvoką verčiant tiesiogiai iš anglų kalbos į lietuvių kalbą¹, gaunami keli nevienareikšmiai vertimo variantai. Pagal vertimo dažnumą yra nurodomi keturi pagrindiniai būdvardžiai:

- **sudėtingas** – turintis daug dalių, komplikuotas²;
- **painus** – sunkiai suprantamas, neaiškus²);
- **kompleksinis** (kompleksas – susijusių daiktų, veiksmų, reiškinių, ypatybių visuma: Gamtinis teritorinis k. Krašto ūkio, pramonės k. Architektūrinis k.²);
- **sudėtinis** – iš dalių sudėtas, sudarytas².

Šio darbo apimtyje sąvoka bus taikoma ir interpretuojama kaip *sudėtinga* ar *sudėtinė* sistema, tačiau ne kaip *kompleksinė* sistema. To priežastis – būdvardžių *painus* ir *kompleksinis* neatitikimas nagrinėjamai sričiai.

Sudėtingos sistemų terminas yra plačiai paplitęs tarp skirtingų disciplinų [LLW13]. Sudėtinga sistema gali būti laikomas internetas [WL13], statybos [Ber03], ekonomika [Art18], ir t. t. Autoriai sąvoką apibrėžia skirtingai:

- tai sistema, sudaryta iš sistemų (sistemų sistema) [BCG⁺08];
- charakterizuojama kaip sistemų rinkinys, kurį lemia sistemos komponentai, jų kiekis ir ryšiai [BPV20];
- sudaryta iš didelio kiekio dalių, kurios sąveikauja netrivialiu būdu [CT19];
- struktūrizuotos sistemos, sudarytos iš daugelio tarpusavyje susietų komponentų, kurie prisideda prie bendros elgsenos, kuri negali būti nuspėjama pagal individualių komponentų elgseną [BD08];
- tai sudėtingi tinklai [WL13];

¹Vertimo metu buvo naudojamas Lietuvių kalbos išteklių informacinės sistemos anglų-lietuvių žodynas. Žr. <http://lkiis.lki.lt/anglu-lietuviu>; tikrinta 2021-04-17.

²Visi vertimai pateikti iš VLKK interneto svetainėje rekomenduojamo dabartinio lietuvių kalbos žodyno tinklalapio. Žr. <http://lkiis.lki.lt/dabartinis>; tikrinta 2020-03-07.

- sistemos, sudarytos iš daugelio nevienodų elementų, susiraizgiusių į netiesinių sąveikų ciklus [MR13].

Taigi, kad sudėtinga sistema negali būti apibrėžta viena bendra taisykle ar apibrėžtimi, dėl prieš tai minėto tarpdiscipliniškumo ir skirtingų kontekstų. Nepaisant to, galima išgryninti esmines sudėtingų sistemų charakteristikas, pasireiškiančias daugumoje termino apibrėžimų.

Dominuojanti charakteristika – sudėtinga sistema sudaryta iš daugiau negu vienos dalies. Tai gali būti bet kokia sudedamoji dalis: komponentas, sistema ar net tokia pati sudėtinga sistema. Tai yra logiška, nes jeigu sistema būtų sudaryta iš vienintelės dalies, kurios negalėtume išskaidyti į daugiau dalių, tai reikštų, kad ji nėra *sudėtinė*. Tai taip pat reikštų, kad išsiaiškinus viską apie vienintelę sistemos sudedamąją dalį, tuo pačiu sužinotume viską ir apie pačią sudėtingą sistemą, o tai pašalintų komplikotumo aspektą ir tokia sistema nebūtų *sudėtinga*. Tokią sudėtingą sistemą galėtume vadinti paprasčiausiai sistema.

Kita svarbi charakteristika – sudėtingos sistemos sudedamųjų dalių sąveika, ryšiai, susietumas. Jeigu sistemą sudarytų n dalių ir jos tarpusavyje nesąveikautų, užtektų išnagrinėti visas n posistemų ir šitaip žinotume viską apie sudėtingą sistemą. Nors tai nebūtinai gali būti trivialus uždavinys, šis uždavinys taptų gerokai sudėtingesnis atsiradus posistemų tarpusavio ryšiams, kurių kiekis galėtų išaugti iki $\frac{n \cdot (n-1)}{2}$.

Ko gero paskutinė charakteristika, kurią verta įtraukti į sudėtingos sistemos apibrėžimą, yra sudedamųjų dalių poveikio darymas bendrai sistemos elgsenai. Posistemų įtakos nebuvimas visai sistemai lemtų, kad nepriklausomai nuo sudedamųjų dalių kiekio, jų tarpusavio ryšių ar specifinių posistemų savybių, visos sudėtingos sistemos elgsena būtų lengvai nuspėjama ar net pasikartojanti. Tai reikštų, kad dvi prieš tai paminėtos sudėtingų sistemų savybės prarastų savo svarbą.

Taigi, sudėtingą sistemą šio darbo kontekste galime apibrėžti kaip daugiakomponentę sistemą, kurios komponentai sąveikauja tarpusavyje ir daro esminę įtaką visos sistemos elgsenai.

1.2. Naujos kartos informacinė sistema

Informacinės sistemos gali būti apibrėžtos kaip bendras įrankis informacijos apdorojimui organizacijoje, įtraukiantis susijusius žmones ir naudojantis informacines technologijas [AGH⁺03]. Informacinės sistemos taip pat gali būti vadinamos tokiomis sistemomis, kurios yra sudarytos iš kompiuterių, programinės įrangos, rolių ir duomenų [RV16].

Istoriškai informacinės sistemos smarkiai kito. Yra išskiriami savybių rinkiniai, kurie įgalina atskirti vienas informacines sistemas nuo kitų. Tokie savybių rinkiniai dar vadinami kartomis

(angl. *generations*). Šiuo metu yra išskiriamos 6 skirtingos kartos, apimančios 1980-2020 metų laikotarpius [PZJ+16]. 1 pav. pateikta lentelė su IS kartomis ir jas nusakančiais atributais.

IS karta	1	2	3	4	5	6
Sritis, į kurią orientuojamasi	Programos	Duomenys	Procesai	Vartotojai	Daiktai	Visuma
Laikmetis	80-ieji	90-ieji	00-ieji	10-ieji	15-ieji	20-ieji
Vyraujanti technologija	MRP	ERP	BPM	HMI	Jutikliai	Prietaisai
Varančiosios technologijos	Duomenų bazės	DBVS, kliento-serverio architektūra	Internetas, servisų architektūra	Semantiniai tinklai, socialinės medijos, debesų kompiuterija, virtuali ir papildyta realybė	Išmanieji daiktai, belaidžių jutiklių tinklai, dideli duomenys, debesų kompiuterija	Atviros ir patikimos platformos, patikima infrastruktūra, suderinamumo servais
Kūrimo aplinka	Lokaliios sistemos		Skaitmeninės sistemos		Kiberfizinės sistemos	
Verslo iššūkiai	Atitikimas	Efektyvumas	Veiksmingumas	Atsparumas	Jautrumas	Proaktyvumas
Organizaciniai iššūkiai	Departamentų palaikymas	Veiklos palaikymas	Tiekimo grandinės palaikymas	Socialinių tinklų palaikymas	Jutiklių tinklų palaikymas	Sąveikos tinklų palaikymas
Technologiniai iššūkiai	Sistemų integravimas			Sistemų suderinamumas		
Duomenų iššūkiai	Struktūrizuoti duomenys	Integruoti duomenys	Dinaminiai duomenys	Realaus laiko duomenys	Nuspėjami duomenys	

1 pav. IS kartos [PZJ+16]

1.3. Naujos kartos informacinės sistemos savybės

Šio darbo kontekste bus nagrinėjamos naujos kartos IS, taigi, šeštoji karta. Kaip teigia autoriai, naujos kartos informacinės sistemos gali būti laikomos tokiomis sistemomis, kurios yra sutelktos į visumą ir yra varomos sukuriant sąveikos tinklus tarp žmonių, duomenų, daiktų ir paslaugų [PZJ+16]. Jų kontekste atsiranda iššūkiai norint pasiekti saugų atvirumą ir gebėjimą sąveikauti su visomis esybėmis [PZJ+16]. Toliau yra pateikiamos ir informacinių sistemų kontekste detalizuojamos tokių sistemų savybės [PZJ+16]:

- **Visur esančios** (angl. *omnipresent*). Tai reiškia, kad informacinės sistemos yra dideliu mastu įgyvendintos arba egzistuojančios skirtingose skaičiavimo ir ne tik platformose. Tai tampa ypač aktualu augant tokių platformų ir prietaisų galingumui [PZJ+16].
- **Modeliais grindžiama** (angl. *model-driven*) **architektūra**. Tokios informacinės sistemos bus grindžiamos pažangiais informacija grindžiamais modeliais, kurie gali kontroliuoti atitinkamos veiklos funkcionalumą [PZJ+16]. Tai sudarytų sąlygas, kuomet pasirenkant vis kitą modelį arba modifikuojant egzistuojantį, funkcinė veiklos elgsena gali būti pakoreguojama arba kontroliuojama nurodant besikeičiančius klientus, aplinką, industriją ar kitus reikalavimus [PZJ+16].
- **Atvirumas** (angl. *openness*). Tai technologinė ir teisinė prieiga prie programinės įrangos

elementų, artefaktų ir išskirstytų sąveikaujančių procesų, grindžiama informacijos dalinimusi su kitais elementais [PZJ⁺16]. Tokio pobūdžio atvirumas, tiek vidinis, tiek išorinis, turi didelę įtaką suderintam organizacijos vidaus įrankių ir aplinkų kūrimui bei išoriniam bendruomenių sutelktumui aplink organizaciją ir atvirai inovacijų rinkai [PZJ⁺16].

- **Dinaminis perkonfigūruojamumas** (angl. *dynamic re-configurability*). Tai yra sistemos gebėjimas funkcionuoti kitokiu būdu ir atlikti tą pačią funkciją arba pasiekti skirtingą rinkinį funkcijų naudojant tą patį arba atnaujintą resursų rinkinį [PZJ⁺16]. Kadangi toks konfigūruojamumas yra dinaminis, reiškia sistema turėtų pasiekti šias savybes realiu laiku.
- **Funkcinių tapatybių daugumas**. Tai reiškia, kad sistemos skirtingose aplinkose ir dalykinės srities kontekstuose bus pajėgios atstovauti daugiau negu vieną funkcinį identitetą [PZJ⁺16].
- **Semantinis ir jutiklinis sąmoningumas**. Ši savybė įgalina atpažinti žemesnio lygio procesų pasikeitimus pasitelkiant jutiklius ir kitus prietaisus. [PZJ⁺16]. Taip pat aukštesnio lygio semnatiinių konstrukcijų atpažinimas [PZJ⁺16].
- **Pažangus samprotavimas ir skaičiavimų lankstumas**. Tokios sistemos turės gebėti samprotauti turint duomenis, kurie yra nepilni, neaiškūs ir išgaunami skirtinguose kontekstuose.

Autoriai taip pat išskiria, kad iš paminėtų savybių kaip neatskiriama savybė išskyla suderinamumas (angl. *interoperability*) [PZJ⁺16].

1.4. Naujos kartos informacinės sistemos abstrakti architektūra

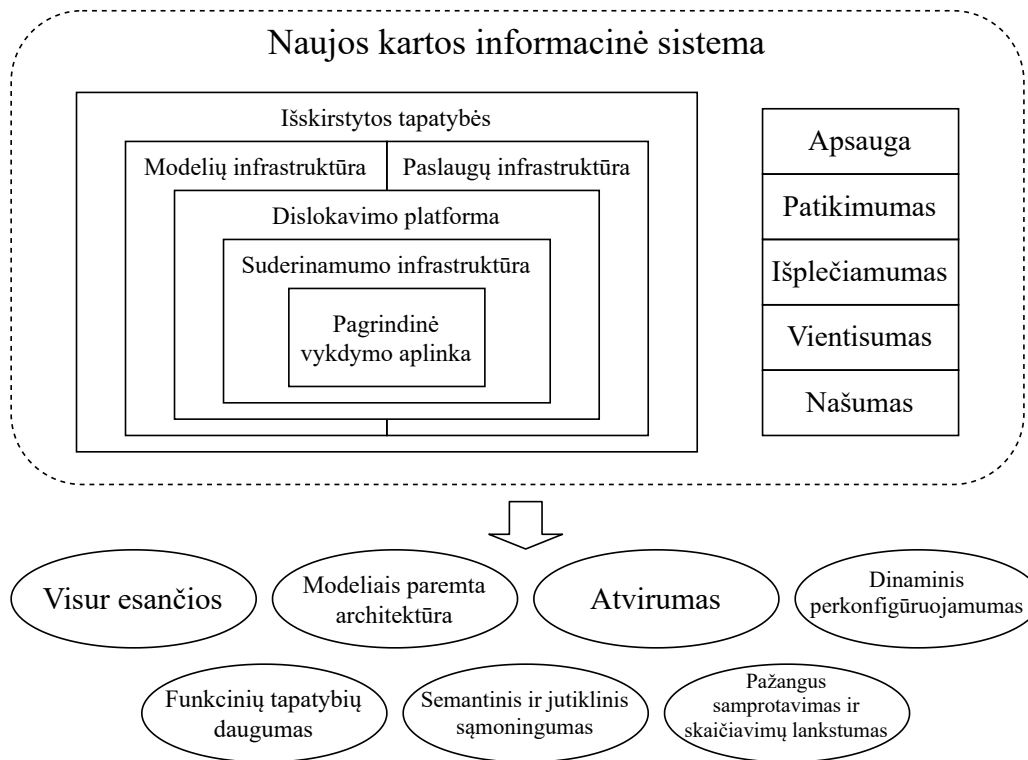
Autorių pasiūlytas naujos kartos informacinių sistemų abstraktus architektūros modelis pateikiamas 2 pav. [PZJ⁺16]. Modelis savyje talpina architektūros komponentų hierarchiją, pagrindinius kokybinius (arba kitaip nefunkcinius) reikalavimus bei prieš tai išvardintas naujos kartos IS savybes.

Autorių pasiūlytame modelyje svarbiausią vaidmenį užima apsaugos (angl. *security*), patikimumo (angl. *trust*), išplečiamumo (angl. *scalability*), vientisumo (angl. *integrity*) bei našumo (angl. *performance*) reikalavimai.

Architektūros komponentai yra šie:

- pagrindinė vykdymo aplinka (angl. *core execution environment*);
- suderinamumo infrastruktūra (angl. *interoperability infrastructure*);
- dislokavimo platforma (angl. *deployment platform*);
- modelių infrastruktūra (angl. *models infrastructure*);

- paslaugų infrastruktūra (angl. *services infrastructure*);
- išskirstytos tapatybės (angl. *distributed identities*).



2 pav. Naujos kartos IS abstraktus architektūros modelis [PZJ⁺16]

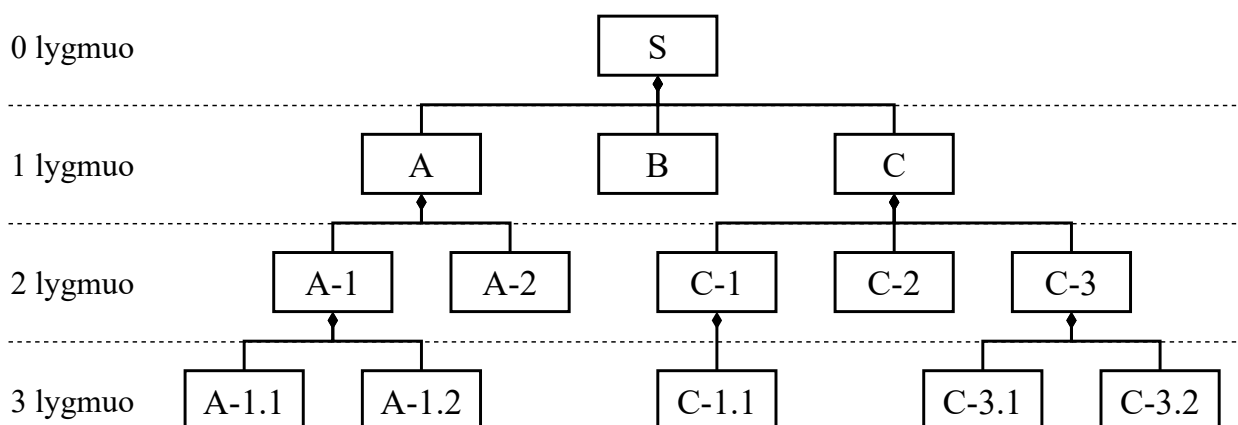
2. Sudėtingų informacinių sistemų dekomponavimas

Šiame skyriuje aptariama dekomponavimo proceso prasmė, svarba ir lokalizavimas sistemos kūrimo gyvavimo cikle. Taip pat pristatoma dekomponavimo būdų formali klasifikacija ir literatūroje sutinkami konkretūs pavyzdžiai. Skyriaus pabaigoje pateikiamas reikalavimų pavyzdinis (angl. *reference*) katalogas.

2.1. Dekomponavimo samprata ir svarba

Kuriant dideles sistemas, dažnai tenka atlikti jų skaidymą į smulkesnes dalis [CHL⁺11]. Viena iš pagrindinių priežasčių yra nurodomas poreikis valdyti ir susitvarkyti su sudėtingumu, dar vadinamas principu „skaldyk ir valdyk“ [Bro01; Koo95; KT11b]. Tikėtina, kad šios smulkesnės dalys taip pat bus sudėtingos ir reikalaus tolimesnio skaidymo į smulkesnes dalis [CHL⁺11]. Tokiu atveju dekomponavimo procedūra gali būti kartojama tol, kol visa sistema tampa išskaidyta į pakankamo dydžio komponentus [CHL⁺11; Koo95].

Dekomponavimo procedūra nėra triviali ir didele dalimi priklauso nuo konteksto, taigi indeterministiška. Egzistuoja įvairūs būdai kaip dekomponuoti sistemas, todėl skirtingų kompetencijų, patirties, išsilavinimo, žinių turintys specialistai tą pačią sistemą dekomponuos skirtingai [CHL⁺11; Koo95; KT11b; SCH15]. Tai rodo, kad egzistuoja alternatyvos, kurių parinkimas yra architekto atsakomybė [SCH15]. Be to, ko gero nėra vieno geriausio būdo tai atlikti [Koo95]. Minėtasis sudėtingumo mažinimas gali būti tiek kiekybinis, tiek kokybinis [KT11b]. Sistemų dekomponavimas turi didžiulę įtaką kokybei ir palaikomumui [Tag11].



3 pav. Sistemos dekompozicija hierarchinio medžio pavidalu

3 pav. pateiktas įsivaizduojamos sistemos dekompozicija hierarchinio medžio pavidalu. Tačiau šią sistemą buvo galima dekomponuoti kitaip, užtikrinant didesnę kiekį posistemų skaičių su

smulkesniu granuliarumu ir daugiau dekomponavimo lygmenų. Be to, pasirinkus kitą dekomponavimo būdą, tai turėtų stiprų poveikį sistemos charakteristikoms – neabejotinai sumažėtų arba padidėtų visos sistemos sudėtingumas.

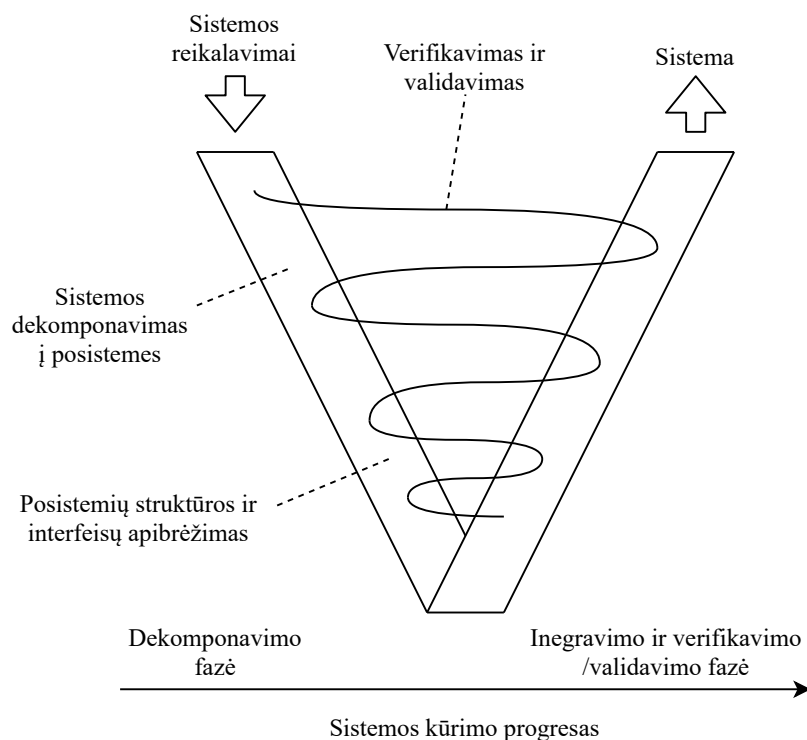
2.2. Dekomponavimo etapai

Kuriant naujas sistemas arba nagrinėjant jau esamas sukurtas, susiduriama su problema: suku suprasti jos visumą [Bro01]. Klasikinis būdas įgauti supratimą apie tokią sistemą yra jos modelavimas trimis etapais [Bro01]:

1. Išskaidyti sistemą į posistemas, apie kurias žinome daugiau, negu apie visą sistemą;
2. Pažymėti sąveiką (integracijas) tarp posistemių;
3. Pažymėti sąveiką su išore, t. y. įeigą ir išeitį bei jų poveikį visai sistemai.

Kiti autoriai pateikia šiuos tris žingsnius, reikalingus dekomponuoti bet kokią sistemą architektūros kūrimo etape, dar vadinamą V modeliu (žr 4 pav.) [KT11b; KT12]:

1. Identifikuoti keliamus reikalavimus sistemos visumai;
2. Dekomponuoti sistemą į posistemas pagal hierarchiją;
3. Apibrėžti posistemių elgseną, struktūrą bei interfeisus tarp jų.



4 pav. Sistemos dekomponavimas V modeliu [KT11b; KT12]

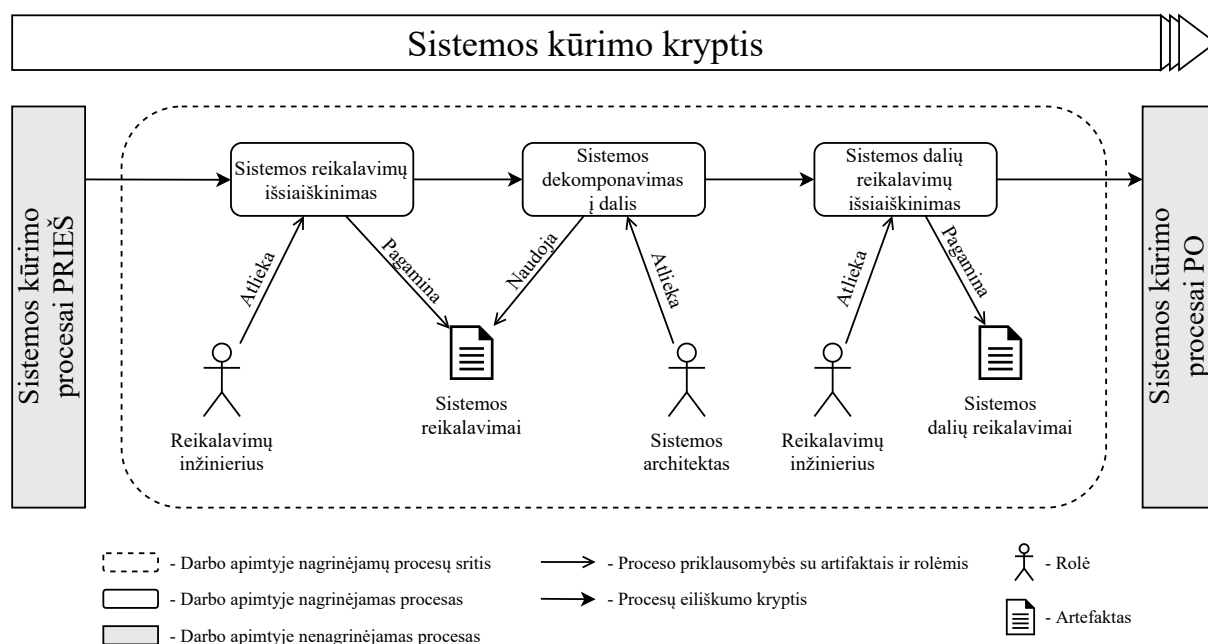
2.3. Dekomponavimo procesas sistemų kūrimo gyvavimo cikle

Dekomponavimo stadija identifikuojama ne visuose sistemų kūrimo gyvavimo cikluose. Dažnai netgi kuriant dideles sistemas šis etapas nevykdomas išreikštiniu būdu arba iš viso yra praleidžiamas dėl specialistų kompetencijų trūkumo, laiko ar biudžeto stokos arba neaiškių sistemos reikalavimų. Kuriant naujos kartos sudėtingas informacines sistemas dekomponavimo procesas turi būti pagrįstas ir metodiškai vykdomas.

Sistemų kūrimo gyvavimo ciklą yra įvairių [Rup10]. Kiekvienos sistemos kūrimo metu gali susidaryti specifinės aplinkybės, dėl kurių gali tekti pritaikyti pasirinktą gyvavimo ciklą ir jo etapų eiliškumą prie esamų poreikių. Dėl šios priežasties darbe bus nagrinėjama tik ta sistemų kūrimo gyvavimo ciklo dalis, kuri yra susijusi su dekomponavimo procesu.

Taip pat svarbu apsibrėžti dekomponavimo proceso priklausomybes su kitais susijusiais ir svarbiais procesais. Prieš dekomponuojant bet kurią sistemą, svarbu turėti tam tikrų žinių apie ją. Priešingu atveju nebus galima atsakyti į klausimus: ką tiksliai dekomponuojame; kuo remiantis dekomponuojame; ką norime pasiekti dekomponuodami sistemą ar jos dalis? Prasmingomis žiniomis apie sistemą galima laikyti konkrečius jai keliamus reikalavimus³.

Taigi, galima teigti, kad sistemos dekomponavimo procesas yra glaudžiai susijęs su reikalavimų išsiaiškinimo (angl. *elicitation*) procesu. Taip yra dėl to, kad reikalavimų išsiaiškinimas gali padėti gauti žinias ir atsakymus į išsikeltus klausimus, reikalingus dekomponavimo procesui. Abiejų procesų priklausomybė pavaizduota 5 pav. [Pen11].



5 pav. Dekomponavimo proceso priklausomybė nuo reikalavimų išsiaiškinimo [Pen11]

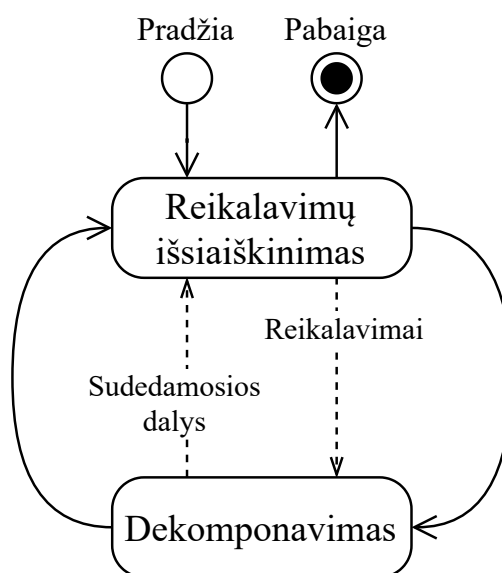
³Šio darbo kontekste terminas „reikalavimas“ yra laikomas sinonimu terminui „savybė“.

Iš pradžių turi būti atliktas visos sistemos reikalavimų išsiaiškinimas. Šį procesą turėtų atlikti kvalifikuotas reikalavimų inžinierius, o proceso metu gautas rezultatas – sistemos reikalavimai. Kaip jau buvo minėta, prieš sistemos reikalavimų surinkimą vykdomi procesai nėra aktualūs. Tačiau svarbu, kad būtų sudarytos sąlygos ir praeiti visi būtini etapai, kurie įgalintų reikalavimų išsiaiškinimą.

Po to yra vykdomas sistemos dekomponavimas į dalis. Šiame etape sistemos architektas pasinaudoja reikalavimų išsiaiškinimo metu parengtu sistemos reikalavimų dokumentu ir, taikydamas vieną arba daugiau alternatyvių dekomponavimo būdų, išskaido sistemą į atskiras smulkesnes dalis.

Atlikus dekomponavimo žingsnį vėl grįžtama prie reikalavimų išsiaiškinimo proceso. Tačiau šį kartą reikalavimų inžinierius analizuoja ne visą sistemą, o kiekvieną sudedamąją dalį atskirai. Kiekvienai iš jų yra pateikiamas analogiškas reikalavimų dokumentas, tačiau sistemos dalių reikalavimai gali skirtis nuo sistemos reikalavimų [Pen11].

Procesą „Sistemos reikalavimų išsiaiškinimas“ galima pavadinti „I-ojo lygmens reikalavimų išsiaiškinimas“, „Sistemos dekomponavimas į dalis“ – „I-ojo lygmens dekomponavimas į dalis“, o „Sistemos dalių reikalavimų išsiaiškinimas“ – „II-ojo lygmens reikalavimų išsiaiškinimas“ ir t. t. Taigi, vietoje terminų „sistema“ arba „sudedamoji dalis“ galima vartoti lygmenų abstrakciją išven-giant techninių terminų.



6 pav. Dekomponavimo ir reikalavimų analizės tarpusavio priklausomybė

Taip pat svarbu atkreipti dėmesį į tai, kad dekomponavimo procesas kartu su reikalavimų išsiaiškinimo procesu vyksta vienas po kito ir kartojami skirtinguose sistemos kūrimo lygmenyse (žr. 6 pav.). Sistemos architektas negali atlikti dekomponavimo neturėdamas sistemos reikalavimų. Atitinkamai ir reikalavimų inžinierius, neturėdamas šių elementų rinkinio, negali vykdyti

žemesniųjų lygmenų sistemos elementų reikalavimų išsiaiškinimo.

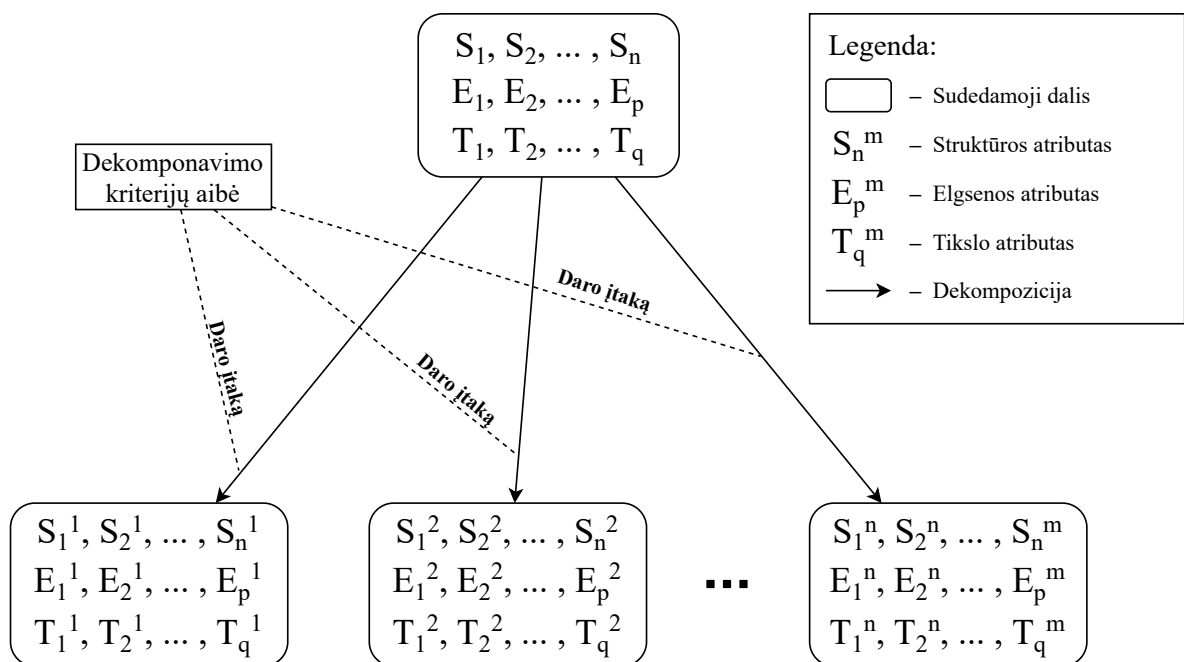
Be to, nors veikla prasideda nuo sistemos reikalavimų surinkimo I-ame lygmenyje, pasirinkus tęsti dekomponavimą tolimesniuose lygmenyse, galutiniame lygmenyje paskutinioji veikla taip pat turėtų būti reikalavimų surinkimas. Taip yra dėl to, kad paskutinio lygmens elementai gali įgauti tam tikras savybes, tačiau jie yra atominiai ir toliau negali būti dekomponuojami į smulkesnes dalis.

Dekomponavimo ir reikalavimų išsiaiškinimo procesų priklausomybės nustatymas yra būtinas siekiant apibrėžti pavyzdinę sistemine dekomponavimo procedūrą.

2.4. Dekomponavimo būdų klasifikacija

Visi įmanomi dekomponavimo būdai gali būti išreikšti dekompozicijos atributų kombinacija. Literatūroje siūlomas toks dekompozicijos atributų klasifikavimas, kuris naudoja trijų atributų kombinacijas [Koo95]:

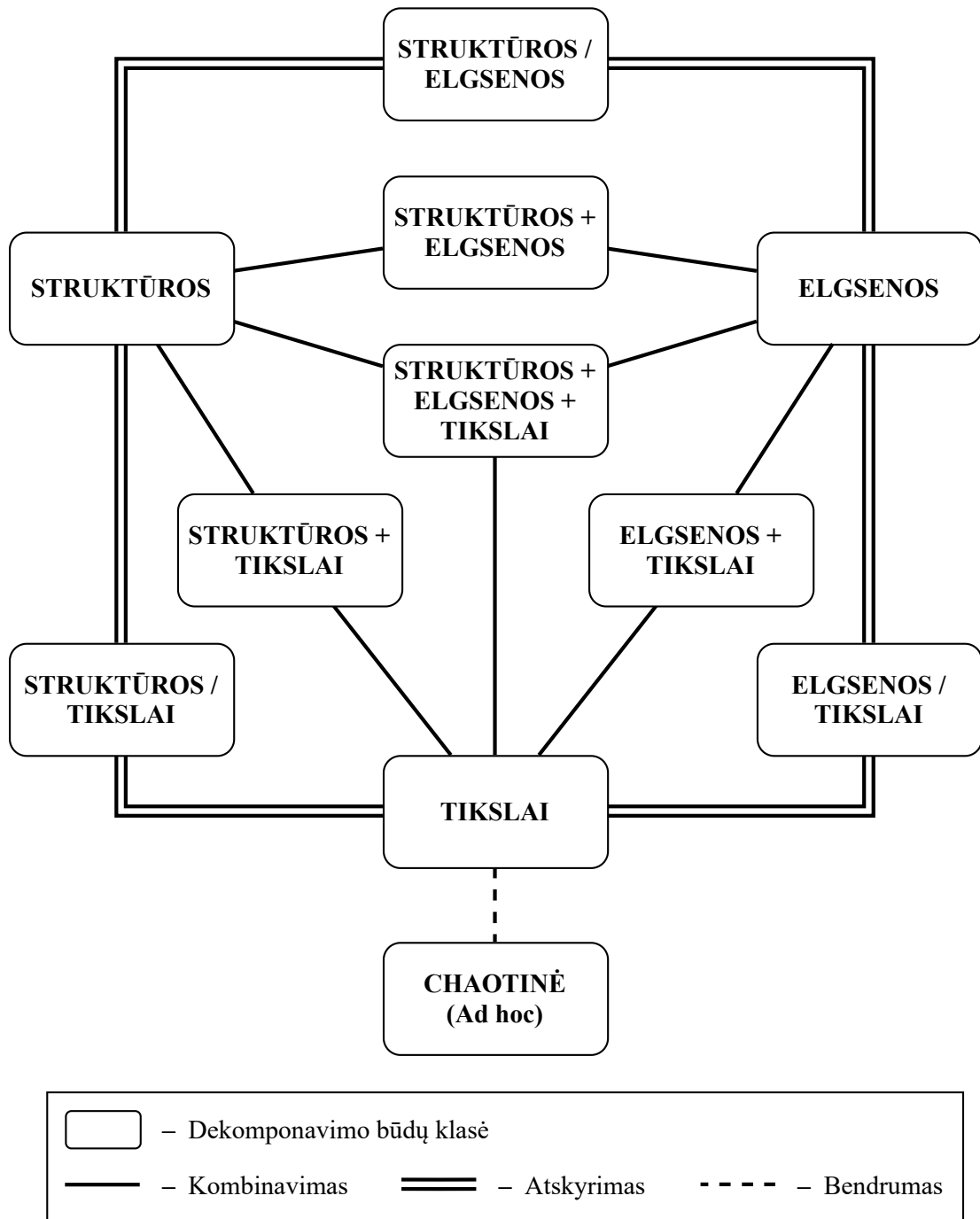
- **Struktūrų.** Tai fiziniai, loginiai ir kiti komponentai, iš kurių yra sudaryta sistema. Pavyzdžiai: elektronika, kompiuteriai, duomenų bazės, programinės funkcijos, kintamieji ir t. t.
- **Elgsenų.** Tai funkcijos, procesai, įvairūs algoritmai. Informacinių sistemų kontekste elgsenas galima traktuoti kaip funkcinis reikalavimus.
- **Tikslų.** Tai visumai būdingos charakteristikos, pavyzdžiui kaina, greitaveika ir t.t. Informacinių sistemų kontekste tikslus galima traktuoti kaip nefunkcinis (kokybinius) reikalavimus.



7 pav. Dekomponavimo kriterijų poveikis dekompozicijai

5 pav. pavaizduotame procese dekompoziciją atlieka sistemos architektas, pasinaudojęs sistemos reikalavimais, kuriuos parengia reikalavimų inžinierius. Tačiau architektas konkrečias sudedamąsias dalis sukuria atsižvelgdamas į dekomponavimo kriterijus (žr. 2.7 poskyrį). 7 pav. pavaizduota, kokią įtaką dekomponavimo kriterijai daro projektavimo spendimams.

Nuo dekomponavimo kriterijų aibės priklauso, kokie struktūriniai atributai atsiras sudedamojoje dalyse, o kartu ir elgsenų bei tikslų atributų priskyrimas joms. Taigi, nuo dekomponavimo kriterijų aibės galiausiai priklausys tai, koks dekomponavimo būdas bus pasirinktas.

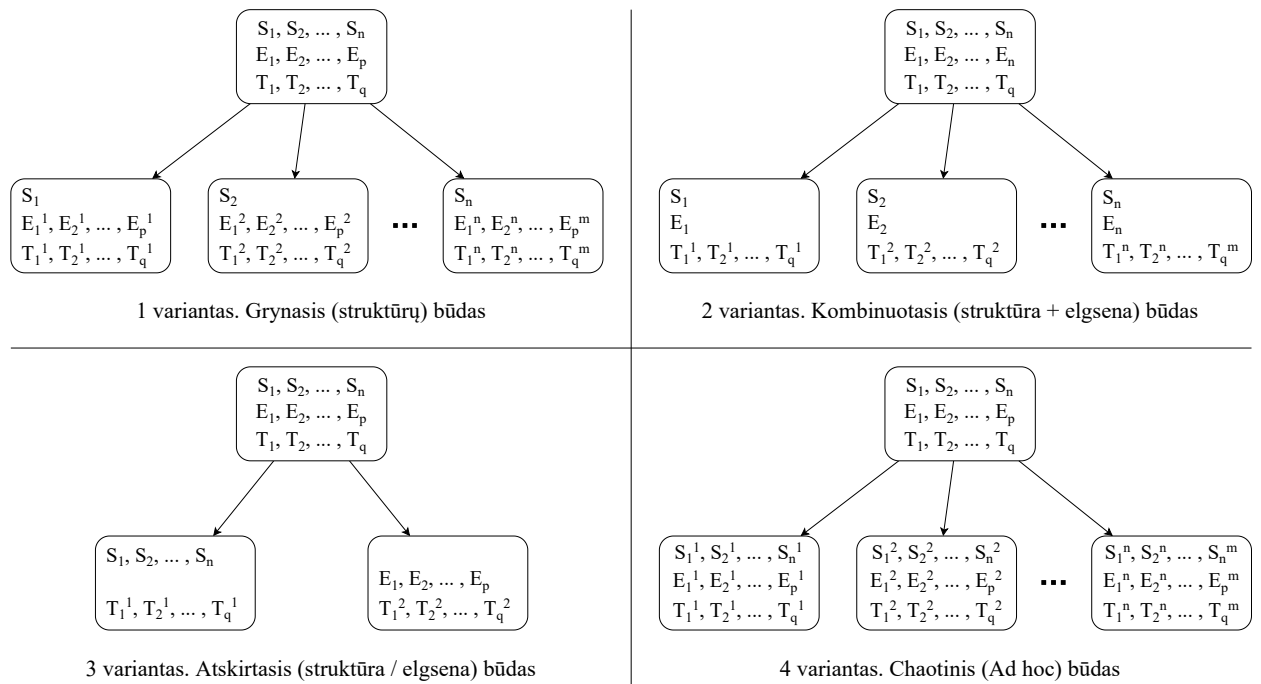


8 pav. Dekomponavimo būdų klasifikacija

Dekomponavimo būdų klasifikacija (žr. 8 pav.) apima 11 klasių, kurias galima sugrupuoti į 4 tipus:

- **Grynosios.** Tai struktūrų, elgsenų ir tikslų dekomponavimo būdų klasės. Šio tipo dekompozicijos pasižymi tuo, kad yra išskirtos pagal vieno tipo atributus taip, kad sudedamojoje dalyje jie būtų unikalūs ir paskirstyti lygiai po vieną. 9 pav. 1 variante pavaizduota struktūrų dekompozicija. Šioje klasėje atsiranda tiek vaikinių projektavimo sprendimų (kitais sudedamųjų dalių), kiek struktūrų yra tėviniame projektavimo sprendime, t. y. kiekviena sudedamoji dalis turi po vieną struktūros S atributą.
- **Kombinuotosios.** Tai struktūrų + elgsenų, struktūrų + tikslų, elgsenų + tikslų ir struktūrų + elgsenų + tikslų dekomponavimo būdų klasės. Šis tipas kombinuoja du arba tris dekomponavimo atributus taip, kad jie nepasikartotų kitose sudedamosiose dalyse. 9 pav. 2 variante pavaizduota struktūrų + elgsenų dekompozicija. Šioje klasėje yra naudojamos 2 atributų – struktūrų S ir elgsenų E poros.
- **Atskirtosios.** Tai struktūrų / elgsenų, struktūrų / tikslų ir elgsenų / tikslų dekomponavimo būdų klasės. Šio tipo dekompozicijose, kitaip negu kitose, atsiranda lygiai dvi sudedamosios dalys, o du skirtingi atributų tipai niekada nebus kartu toje pačioje sudedamojoje dalyje. 9 pav. 3 variante pavaizduota struktūrų / elgsenų dekompozicija. Šioje klasėje atsiranda lygiai 2 sudedamosios dalys, nes į vieną perkeliama visi struktūrų, o į kitą visi elgsenų rodikliai.
- **Chaotinės.** Šis tipas apima vienintelę dekomponavimo būdų klasę, kuri vadinama chaotine arba *ad hoc* ir pavaizduota 9 pav. 4 variante. Šis atributų priskyrimas sudedamosioms dalims nėra sisteminis ir autoriaus įvardinamas kaip nemokslinis [Koo95]. Taip yra todėl, kad bet kokie atributai gali būti priskirti visoms sudedamosioms dalims.

Šio darbo apimtyje bus nagrinėjami tik tie dekomponavimo būdai, turintys tokius struktūrinius atributus, kurie nepasikartoja sudedamosiose dalyse. Šias sąlygas atitinka penkios dekomponavimo būdų klasės: grynoji struktūrinė, kombinuotoji struktūros + elgsenos, kombinuotoji struktūros + tikslai, kombinuotoji struktūros + elgsenos + tikslai ir elgsenos / tikslai.



9 pav. Dekomponavimo būdų klasių pavyzdžiai

2.5. Dekomponavimo būdai

Atliekant literatūros analizę užsienio autorių darbuose apie sistemų dekomponavimą, buvo atrinktas rinkinys skirtingų dekomponavimo būdų pavyzdžių su kiekvienos apibūdinimais, privalumais ir trūkumais. Gauti analizės rezultatai pristatomi 2.5.1 – 2.5.7 skirsniuose.

2.5.1. Probleminiais freimais grindžiama dekompozicija

Probleminiai freimai (angl. *problem frames*) gali būti apibrėžiami kaip sistemiškas problemų dekomponavimas, įgalinantis susieti reikalavimus, dalykinės srities problemas ir sistemos specifikaciją [LBJ⁺04].

Probleminių freimų dekomponavimo metodas kalba apie tai, kad į sistemą reikia žiūrėti kaip į darinį, sprendžiantį kokią nors problemą. Toliau yra numatyta procedūra dekomponavimui – ši problema vis skaidoma į mažesnes atskiras dalis iki kol pasiekiamas žinomų probleminių freimų mastelis [CKL12]. Tai įgalina identifikuoti visas kuriamos sistemos sprendžiamą problemą sudarančias antrines problemas [CKL12].

Tinkamai skaidant problemas į smulkesnes problemas atsiranda įvairių privalumų, jeigu mažesnes problemas sprendžiame jas izoliuodami ir tada su jomis atliekame priešingą dekompozicijai darbą – kompoziciją (angl. *composition*) [LBJ⁺04]. Tokie privalumai yra [LBJ⁺04]:

- išplečiamumas (angl. *scalability*) – dėl to, kad dirbama paprastesnių problemų lygmenyje;

- atsekamumas (angl. *traceability*) – problemos tiesiogiai susiejamos su jų sprendimais;
- paprastesnė sistemos evoliucija (angl. *evolution*) – pasikeitimai dekomponuotose problemose įgyvendinami modifikuojant atitinkamus problemų sprendimus.

Šis metodas sumažina projektavimo pastangas pakartotinai panaudojant turimas žinias [CKL12]. Be to, analitikai ir inžinieriai gali dirbti lygiagrečiai, o tai yra naudinga, nes potencialiai sumažinama sistemos kūrimo trukmė ir kaina [CKL12].

Tačiau yra ir papildomų trūkumų – norint sėkmingai taikyti metodą reikia turėti stiprias dalykinės srities žinias, analitinį ir žinių kapitalą [CKL12]. Be to, probleminiai freimai tinkami, jeigu problema yra standartinė, turinti fiksuotą aprašą, tačiau realybėje problemos nuolat kinta ir šį kitimą reikia prižiūrėti [RHJ⁺04].

Tarpusavyje derinant NFR karkasą ir probleminius freimus galima sumodeliuoti sistemos kokybės atributus ir atpažinti potencialiai netinkamas ar nepakankamas dalykinės srities prielaidas [DLK11]. Šiuo būdu yra sudaromos galimybės modeliuoti dekomponuotą sistemą sukuriant įvairias alternatyvas, kurios užtikrintų norimas sistemos savybes [DLK11].

Taigi, naudojant probleminius freimus galima pasiekti, kad būtų užtikrinama bet kokia sistemos savybė, jeigu tik egzistuoja gerai žinomi probleminiai freimai, atitinkantys norimą savybę arba siekiamą išspręsti problemą. To pavyzdys – apsaugos užtikrinimas [HHS06].

Trūkumai. Papildomos pastangos, kai kinta dekomponuotos problemos, taikymui reikalingos dalykinės srities žinios, analitinis ir žinių kapitalas.

Privalumai. Įgalina paprastesnį plečiamumą, atsekamumą, sistemos evoliucijos valdymą, galima užtikrinti pasirinktas sistemos savybes.

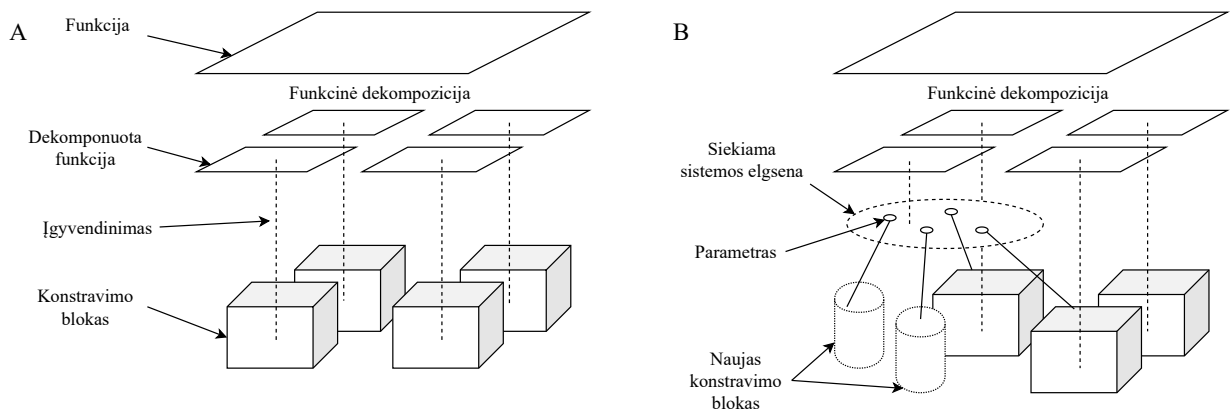
2.5.2. Funkcionalumu grindžiama dekompozicija

Funkcinė (angl. *functional*) dekompozicija – tai kai sistema dekomponuojama į komponentus remiantis funkciniais reikalavimais [Löw19]. Pavyzdžiui, jeigu sistema turi vykdyti N kiekių funkcinį operacijų, tai atitinkamai turės būti sukurtas N kiekis servisų, įgyvendinančių būtent šias funkcines operacijas [Löw19].

Kai kurie autoriai pateikia rekursinį funkcinės dekompozicijos modelį, kuomet remiantis funkciniais sistemos reikalavimais yra dekomponuojamos funkcijos, kol to daryti daugiau nebe galima [KT12]. Tam tikros dekomponuotos funkcinės dalys yra pakartotinai naudojamos konstravimo blokų pavidalu (angl. *building blocks*), kurie programinės įrangos kūrimo kontekste vadinami moduliais (angl. *modules*) arba vienetais (angl. *units*) [KT12]. Neturint šių konstravimo blokų jie

turi būti sukurti [KT11a; KT12]. Funkcinė dekompozicija dviem variantais, kai a – konstravimo blokai yra žinomi ir kai b – konstravimo blokai nėra žinomi, yra pavaizduota 10 pav.

Funkcinės dekompozicijos metodas yra intuityvus ir nesudėtingas bei įgalina atrasti paslėptus arba neapibrėžtus funkcinius reikalavimus [Löw19]. Tačiau kritikai teigia, kad toks metodas yra netinkamas dėl servisų suporavimo su reikalavimais [Löw19]. Reikalavimų kaita yra neišvengiama kuriant sistemas ir bet koks reikalavimų pasikeitimas turės atsispindėti atitinkamuose programiniuose servisuose [Löw19].



10 pav. funkcinės dekompozicijos metodas: a) turint konstravimo blokus ir b) neturint konstravimo blokų [KT12]

Be to, tiesiogiai poruojant servisuose ir funkcionalumus, žemiausiame dekompozicijos lygmenyje atsiras milžiniškas kiekis servisų, kurių funkcionalumas tikėtina bus dubliuojamas kitose sistemos dalyse [Löw19]. Šis servisų sproginimas sukelia neproporcingai aukštą integravimo ir testavimo kainą bei sistemos bendrą sudėtingumo laipsnį [Löw19].

Trūkumai. Išauga sistemos sudėtingumas, padažnėja funkcionalumo dubliavimas, apsunkintas integravimas, testavimas, nepritaikyta dažnai kintantiems reikalavimams.

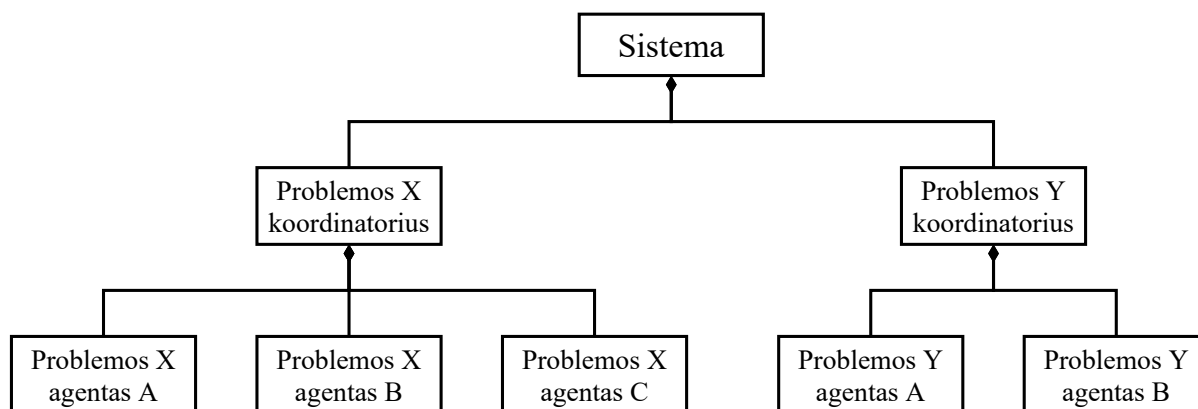
Privalumai. Intuityvi, nesudėtinga dekomponavimo proceso prasme, galima sistemiškai atrasti užslėptą arba neapibrėžtą sistemos funkcionalumą.

2.5.3. Agentais grindžiama dekompozicija

Agentais (angl. *agent*) grindžiama dekompozicija – tai į optimizaciją ir našumą sutelktas dekomponavimo būdas [HPD08]. Optimizacijos ir našumo privalumai atsiranda tada, kai sistemą išskaidome į savarankiškus agentus, atsakingus už konkrečią veiklą. Hierarchinis modelis pateiktas 11 pav. Diagramoje galima atpažinti šeimininko-vergo (angl. *master-slave*) principą, kur valdovas yra koordinadorius, o vergas – už tam tikrą veiklą atsakingas agentas.

Koordinatorius sprendžia kada kreiptis į agentus ir teikia jiems reikalingus duomenis, kuriuos apdorojęs agentas grąžina atgal koordinatoriui (bet nebūtinai). Koordinatorius neteikia agentams instrukcijų, kaip vykdyti jų darbą, todėl agentai yra autonominiai [HPD08].

Jeigu visa sistema būtų dislokuota viename kompiuteryje, būtų prarandamas konfidencialumas, nes koordinatoriams turėtų būtų suteikiama visa, net ir jautri informacija [HPD08]. Išskirstant agentus po skirtingas mašinas problemos nebėra, nes koordinatoriai mato agentus kaip juodas dėžes.



11 pav. Agentų dekompozicijos pavyzdys

Be to, sistemai esant viename kompiuteryje, prarandamas efektyvumas dėl komunikacijos kanalų ir skaičiavimų apkrovos [HPD08]. Paskirstant agentus po skirtingas mašinas padidėja bendras sistemos skaičiavimų galingumo potencialas, skaičiavimai ir duomenų perdavimas gali būti vykdomi lygiagrečiai [HPD08]. Galiausiai, esant išskirstymui sistema išvengia pažeidžiamų taškų (angl. *single point of failure*) [HPD08]. Sutrikus koordinatoriaus veiklai, atitinkamai sutrinka ir visa problemos sprendimo sritis, tačiau esant vieno ar kelių agentų trikdžiams, kiti agentai ir koordinatorius gali toliau sėkmingai vykdyti savo veiklą [HPD08].

Trūkumai. Esant dideliame kiekiui agentų, išauga komponentų komunikavimo kaštai ir koordinatoriai tampa apkrauti, o centralizuotoje sistemoje prarandami pagrindiniai dekomponavimo būdo privalumai.

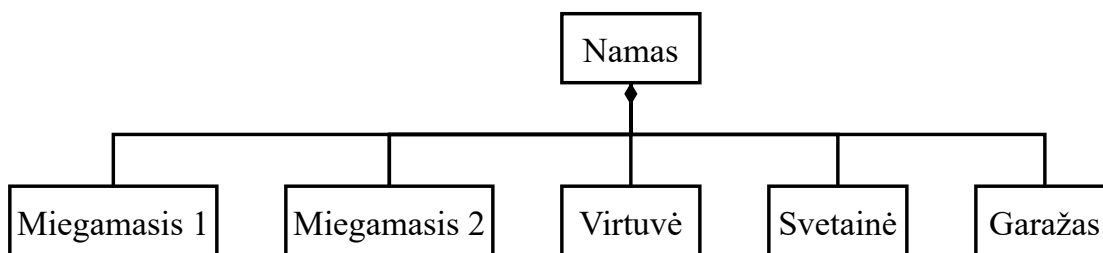
Privalumai. Užtikrinamas konfidencialumas, išskirstytų skaičiavimų galimybė, padidėjęs sistemos tvarumas.

2.5.4. Dalykine sritimi grindžiama dekompozicija

Dalykinės srities (angl. *domain*) dekompozicijos metodas – tai kuomet sistema suskaidoma į dalis pagal verslo sritis [Lów19]. Pavyzdys – namo analogija programinei įrangai (žr. 12 pav.). Sistema suskaidoma į virtuvę, miegamąjį 1, miegamąjį 2, garažą ir svetainę. Kiekvienas kambarys

turės savo funkcijas – virtuvėje maisto gamyba, miegamajame miegas ir t. t. Kiekvienas funkcionalumas gali būti priskirtas sričiai.

Tačiau iš to kyla potencialios problemos. Nors *Miegamasis 1* ir *Miegamasis 2* yra skirtingi kambariai, miego funkcionalumą teks atkartoti abiejuose kambariuose [Löw19]. Susijęs funkcionalumas taip pat bus išskirstytas per skirtingus kambarius, taigi komponentai glaudžiai susiję, o tai iššaukia padidėjusį sistemos sudėtingumą [Löw19]. Be to, gali atsirasti funkcionalumų, kurių negalima logiškai priskirti nei vienai sričiai. Naudojant dalykinės srities dekomponavimo būdą beveik neįmanoma dislokuoti vieno funkcionalumo izoliuotai [Löw19].



12 pav. Dalykinės srities dekompozicijos pavyzdys

Testavimas taip pat tampa sudėtingesnis. Dėl didelio komponentų sukibimo ir sudėtingumo programuotojai galėtų atlikti nebent vienetų testavimą [Löw19]. Regresinis testavimas taptų sunkiai įmanomas dėl daugelio komponentų tarpusavio jungčių [Löw19].

Trūkumai. Dubliuojamas funkcionalumas. Susijęs funkcionalumas išskirstomas per skirtingas sritis. Padidėjęs komponentų sukibimas ir sudėtingumas. Regresinis sistemos testavimas yra komplikuoatas.

Privalumai. Tokia dekompozicija gali būti naudinga bendraujant su verslu, kuris gerai išmano dalykinę sritį.

2.5.5. Kintamumu grindžiama dekompozicija

Kintamumu (angl. *volatility*) grindžiama dekompozicija identifikuoja potencialias sistemos sritis, kurios dažnai keičiasi ir enkapsuliuoja jas servisų arba komponentų pavidalu. Toks suskaidymas pagal sistemos kitimo sritis įgalina sumažinti sistemos sudėtingumą ir užtikrina paprastesnį palaikymą [Löw19].

Be to, dekomponavus sistemą pagal kintamumo sritis regresiniai testai tampa lengvai realizuojami. Taip yra todėl, kad kiekviena posistemė laiko savyje pasikeitimus ir užtenka ištestuoti tik konkrečią posistemę, neličiant kitų sistemos dalių [Löw19].

Tačiau šis dekomponavimo būdas yra mažiau intuityvus ir įprastai užima daugiau laiko nei kiti

metodai dėl komunikacijos su suinteresuotomis šalimis ir išsamesnės sistemos reikalavimų analizės [Löv19].

Trūkumai. Dekompozicija nėra intuityvi, užima daugiau laiko ir analizės pastangų.

Privalumai. Sistema efektyviai pritaikoma dažniems pokyčiams, supaprastinamas palaikymas, regresinių testų kūrimas.

2.5.6. Servisais grindžiama dekompozicija

Servisų (angl. *service*) dekompozicija remiasi sistemos aprašu ir žinynu (angl. *manual*) [CHL⁺11]. Kiekvienas žinyno skyrius atitinka pirmąjį dekomponavimo lygmenį, poskyris antrąjį lygmenį ir t. t. [CHL⁺11]. Tai yra servisų inžinierių perspektyva, per kurią yra tvarkoma ir palaikoma sistema [SCH15]. Taigi, sistema išskaidoma į dviejų tipų dalis: į tas posistemas, kurios reikalaujama periodinio keitimo bei palaikymo arba tokias, kurias reikėtų keisti ar prižiūrėti kuo mažiau. Be to, posistemės, kurioms yra reikalinga priežiūra, turėtų pasižymėti savybe, kad būtų lengvai pašalinamos ar pakeičiamos [CHL⁺11].

Trūkumai. Sunku pritaikyti naujai kuriamoms sistemoms. Nenumatomos servisų tarpusavio sąveikos.

Privalumai. Dekomponavimo būdas tinkamas tokioms sistemoms, kurioms reikia intensyvaus palaikymo ir konfigūracijų keitimo.

2.5.7. Surinkimu grindžiama dekompozicija

Surinkimo (angl. *assembly*), arba dar kitaip fizinė dekompozicija – kada didelės sistemos dalys fiziškai atskiriamos nuo sistemos [CHL⁺11]. Įprasta, kad ją naudoja gamybos inžinierių, kurių atsakomybė yra surinkti galutinę sistemą [SCH15]. Naudojant šią dekompoziciją yra siekiama suprojektuoti sistemą ir jos posistemas lengvam surinkimui [SCH15]. Gamybos linijos nuoseklumui užtikrinti, kiekvienos posistemės surinkimo laikas turėtų trukti beveik tiek pat [SCH15]. Pirmojo dekompozicijos lygmens posistemės turėtų būti surenkamos lengviausiai, identifikuojant tiesiogines jungtis tarp jų [SCH15]. Dekomponavimo metu taip pat svarbu atkreipti dėmesį į granuliarumą bei sistemos ir posisteminių dalių dydžių santykius [CHL⁺11].

Trūkumai. Taikant dekomponavimo būdą posistemės nebūtinai turės savo tapatumą, t. y. jos savyje turės komponentus, kurie atliks skirtingas funkcijas [SCH15]. Taip pat šis dekomponavimo būdas sunkiai pritaikoma grynai programinei įrangai, neturinčiai fizinių komponentų ir tokiai sistemai, kuri dar nėra sukurta.

Privalumai. Gerai pritaikoma, kai kalbama apie fizinę ir jau egzistuojančią sistemą. Metodus taip pat pabrėžia architektūros struktūrinius aspektus [SCH15]. Surinkimo dekomponavimo būdas tiktų kaip hibridinio metodo dalis, kuomet informacinės sistemos programinė dalis dekomponuojama naudojant vieną būdą, o fizinė (aparatinė, techninė įranga angl. *hardware*) – surinkimo būdu.

2.6. Sistemos reikalavimų pavyzdinis katalogas

Norint atlikti tinkamą sistemos arba bet kurio jos elemento dekompoziciją, sistemos architektui reikia žinoti norimas savybes (kitais reikalavimus) (žr. 5 pav.). Atlikus dekompoziciją, kuriama sistema turėtų tenkinti jai iškeltus reikalavimus. Savybių gali būti daug ir įvairių, jos priklauso nuo sistemos ypatumų ir, neturint formalizuotų jų identifikavimo metodų, nuo reikalavimų inžinieriaus kompetencijų.

Be to, ne visos norimos sistemos savybės gali būti pasiektos kartu – kai kurios gali prieštarauti vienos kitoms. Jeigu ir nėra prieštaringų reikalavimų, visų jų įgyvendinimo kaštai gali būti per dideli. Galiausiai reikalavimai, keliami sistemos elementui, gali skirtis nuo reikalavimų, keliamų elemento sudedamosioms dalims.

2.4 skyriuje pristatyti dekomponavimo būdų klasių atributai (elgsenos ir tikslai) dekomponavimo proceso metu tampa konkrečiomis savybėmis (kitais tariant reikalavimais). Toliau bus siekiama nustatyti, kaip skirtingus reikalavimus galima paskirstyti į konkrečių atributų aibes. Literatūroje [Pen11] skiriamos šios savybių kategorijos, o skliaustuose nurodoma atributų aibė, kuriai priskiriami konkretūs reikalavimai:

- funkciniai reikalavimai (elgsenos);
- direktyviniai reikalavimai (tikslai);
- kokybiniai reikalavimai (tikslai).

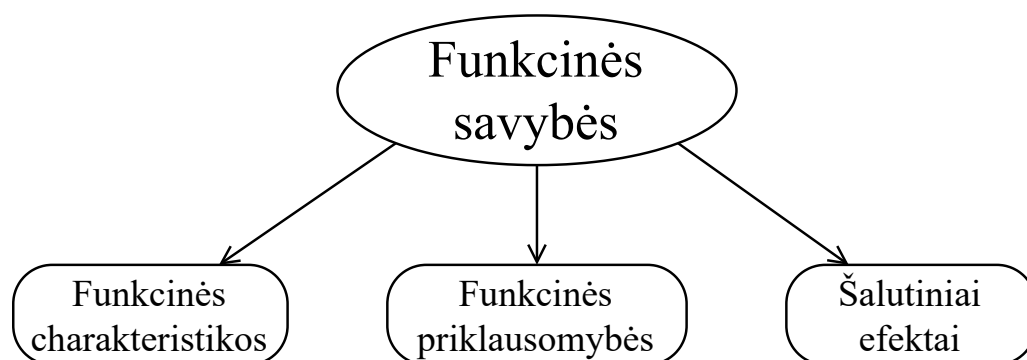
2.6.1. Funkcinės savybės

Yra išskiriamos 3 funkcinų savybių sub-kategorijos (žr. 13 pav.):

- *Funkcinės charakteristikos.* Šios savybės nusako funkcinės sistemos charakteristikas iš sistemos naudotojo požiūrio taško.
- *Funkcinės priklausomybės.* Šios savybės nusako, kad servisas ar funkcija gali būti vykdoma tik tuo atveju, jeigu kitas servisas atliko darbą arba laukia duomenų iš duotojo serviso. Taigi,

nusakomas būtinas eiliškumas bei priklausomybės. Šias savybės apibrėžti dekomponuojant sistemą nuo viršaus iki apačios nėra triviale, nes priklausomybių detalės tampa aiškesnės turint žinių apie sistemą.

- *Šalutiniai efektai arba nepageidaujama funkcijų sąveika.* Tai sąveika, atsirandanti dėl funkcionalumų daromos įtakos kitiems funkcionalumams, kuri nebuvo numatyta. Nors nenumatyta sąveika gali būti teigiama, dažniausiai ji turi nepageidaujamą poveikį sistemai. Pagrindinis būdas įsitikinti ar tokių sąveikų sistemoje nėra – atlikti analizę.



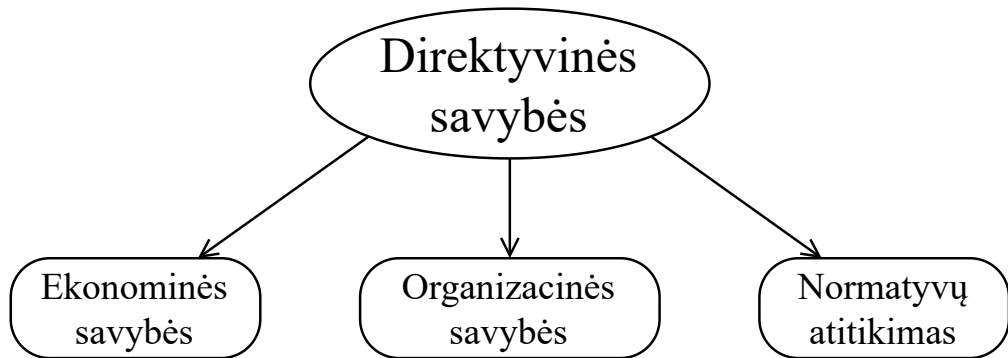
13 pav. Funkcinės savybės

2.6.2. Direktyvinės savybės

Direktyvinės savybės nusako įstatymus, standartus, licencijas, patentus ir verslo taisykles. Tai savybės, veikiančios sistemos dekompoziciją iš išorės. Kai kurios savybės daro poveikį sistemai tiesiogiai ir negali būti keičiami. Kitos savybės daro poveikį netiesiogiai ir gali būti modifikuojamos, siekiant pritaikyti juos sistemai. Yra išskiriamos trys direktyvinių savybių sub-kategorijos (žr. 14 pav.):

- *Ekonominės savybės.* Tai pakartotinis panaudojimas, paklausos valdymas, kaštų modeliai. Paklausos valdymas apima padėtį rinkoje ir strateginius tikslus. Pakartotiniam panaudojimui turi būti atlikta analizė, identifikuojanti komponentų variacijas ir numatanti pasikeitimus bei galimus adaptavimo būdus.
- *Organizacinės savybės.* Tai kuriamos sistemos kontekste esanti infrastruktūra ir Konvėjaus dėsnis, kuris teigia, jog kuriamos sistemos imituoja tas sistemas kuriančių organizacijų komunikacinę struktūrą [HG99].
- *Normatyvų atitikimas.* Tai įstatymai, standartai, licencijos, patentai ir sertifikatai. Šių savybių įgyvendinimo svarba yra visuotinai suprantama, todėl jų nesistengiama apeiti ir jų

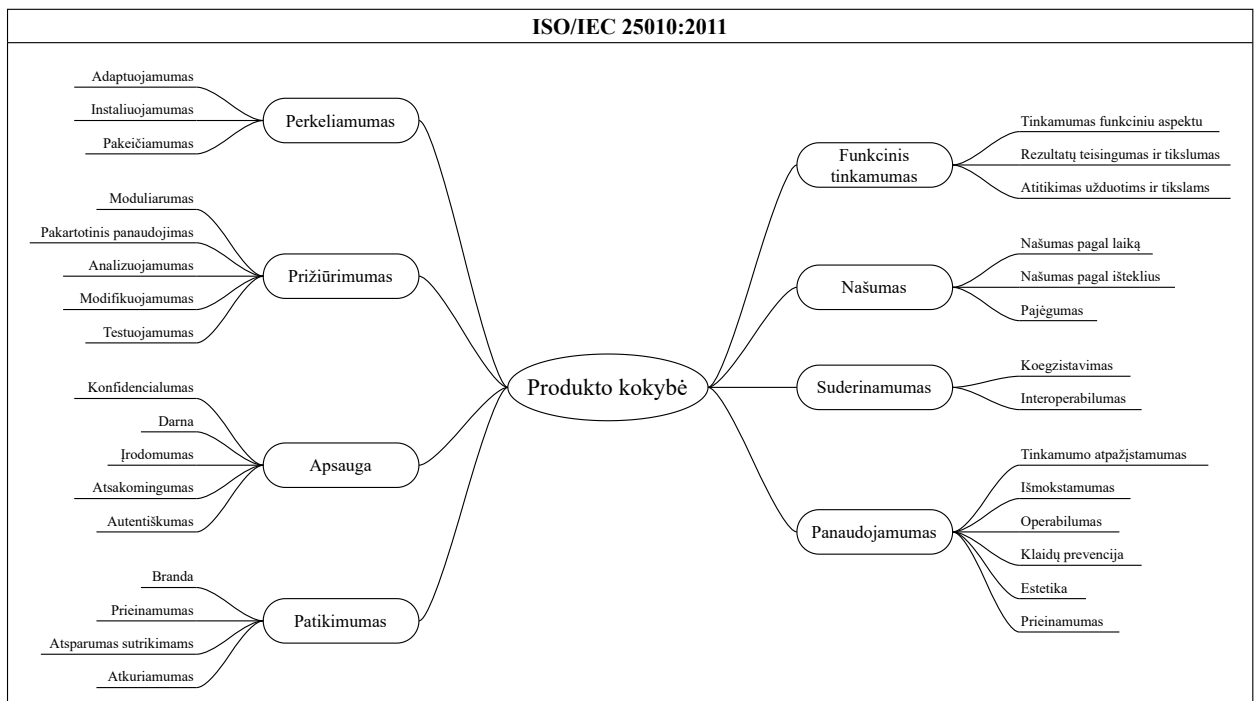
įgyvendinimas yra būtinas.



14 pav. Direktyvinės savybės

2.6.3. Kokybinės savybės

Kokybiniai reikalavimai (dar vadinami nefunkciniais) pavaizduoti 15 pav. Šiame darbe naudojamas kokybinių reikalavimų rinkinys yra apibrėžtas standarte ISO/IEC 25010:2011 [IEC11]. Šiame standarte yra išskiriamos aštuonios nefunkcinių reikalavimų kategorijos, kurių kiekviena turi po dar kelias sub-kategorijas, kurių iš viso yra 31.



15 pav. Kokybinės savybės

2.7. Dekomponavimo kriterijai

Kaip jau aptarta 2.4 poskyryje, dekomponavimo kriterijai apsprendžia dekomponavimo būdo pasirinkimą. Dekomponavimo kriterijai dažnai yra ribojimai arba būtinosios sąlygos, kurios atsiranda dėl kuriamos sistemos ypatumų. Pavyzdžiui, kuriant naujos kartos sudėtingas informacines sistemas, dekomponavimo kriterijų aibė skirsis nuo kompiuterių operacinių sistemų kriterijų aibės.

Tačiau šio darbo kontekste tokios kriterijų aibės nebus nagrinėjamos, kadangi darbo tikslas yra ištirti dekomponavimo būdų poveikį sistemoms, o ne kriterijų poveikį joms.

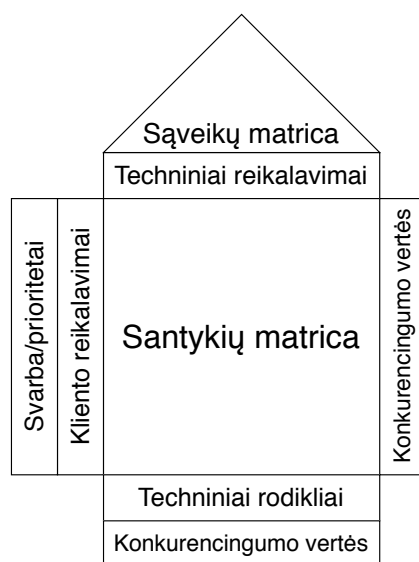
3. Dekomponavimo kortų metodas

Šiame skyriuje glaustai apžvelgiama originali kokybės funkcijų sklaidos metodika ir detaliai aprašomas autoriaus sukurtas informacinių sistemų dekomponavimo kortų (toliau – DK) metodas pasirinktam dekomponavimo būdui įgyvendinti, analizuoti ir palyginti su alternatyvomis. Skyriaus pabaigoje apžvelgiami DK metodo ir jo pagrindu sukurtos priemonės ribojimai ir plėtros kryptys.

3.1. Originali kokybės funkcijų sklaidos metodika

Reikalavimų inžinerijoje yra įvairių metodikų, skirtų sistemos reikalavimų išsiaiškinimui. Vienas tokių būdų yra kokybės funkcijų sklaidos (angl. *quality function deployment*, toliau – KFS) metodika. Joje kaip esminis įrankis naudojama kokybės korta, kitaip dar vadinama kokybės nameliu (angl. *house of qualities*) [TYW99].

Metodika yra laiko patikrinta ir patikima. Svarbiausi metodikos vystymo etapai yra šie: 1972 metais pradėta taikyti Japonijos automobilių pramonėje [CW02; RMC98], iki XX a. pabaigos paplinta visame pasaulyje bei pradėdama taikyti programų sistemų kūrimo srityje [CW02] ir diegiama tarptautinėse kompanijose, tokiose kaip IBM [CW02]. 2015 metais KFS metodika standartizuota ISO standartų šeimoje⁴ [MB17].



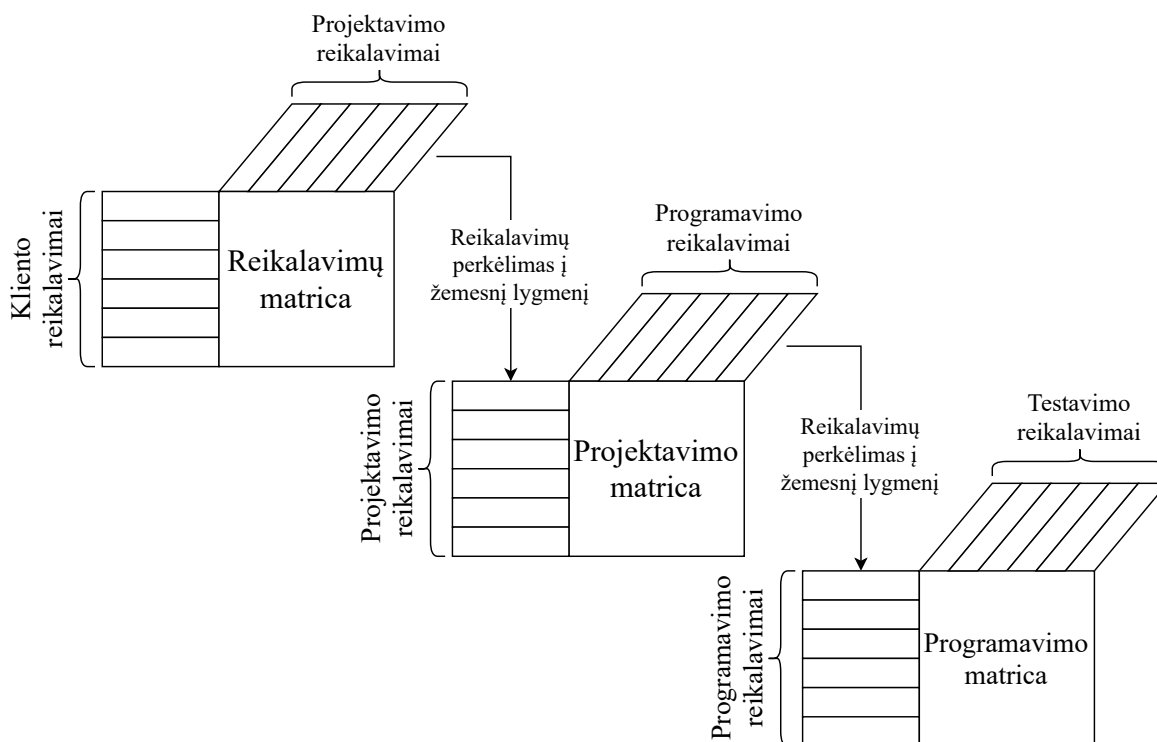
16 pav. KFS kortos apibendrinta versija

KFS tikslas – identifikuoti naudotoją bei transformuoti jo poreikius į tolimesnes įgyvendinamas produkto inžinerines charakteristikas [RMC98; TYW99]. Tam yra naudojamos kortos. Originali KFS korta pateikta 16 pav. Ją sudaro šios pagrindinės dalys: kliento reikalavimai, reikalavimų

⁴Žr. <https://www.iso.org/standard/62626.html>; tikrinta 2021-01-16.

svarba (prioritetai), techniniai reikalavimai, techninių reikalavimų sąveikų matrica, kliento ir techninių reikalavimų ryšių matrica, techniniai rodikliai, konkurencingumo vertės (atitinkamai kliento reikalavimų ir techninių reikalavimų ašyse) [TYW99].

Tačiau 16 pav. pavaizduota karta susieja kliento reikalavimus su techniniais reikalavimais ir ji netinka, jeigu norėtume susieti techninius reikalavimus su įgyvendinamaisiais reikalavimais. Taigi, nėra aišku, kaip pereiti prie žemesniojo lygmens reikalavimų. KFS metodas padengia šį aspektą kortų perkėlimo į žemesnį lygmenį principu (žr. 17 pav.) [HC88; TYW99].



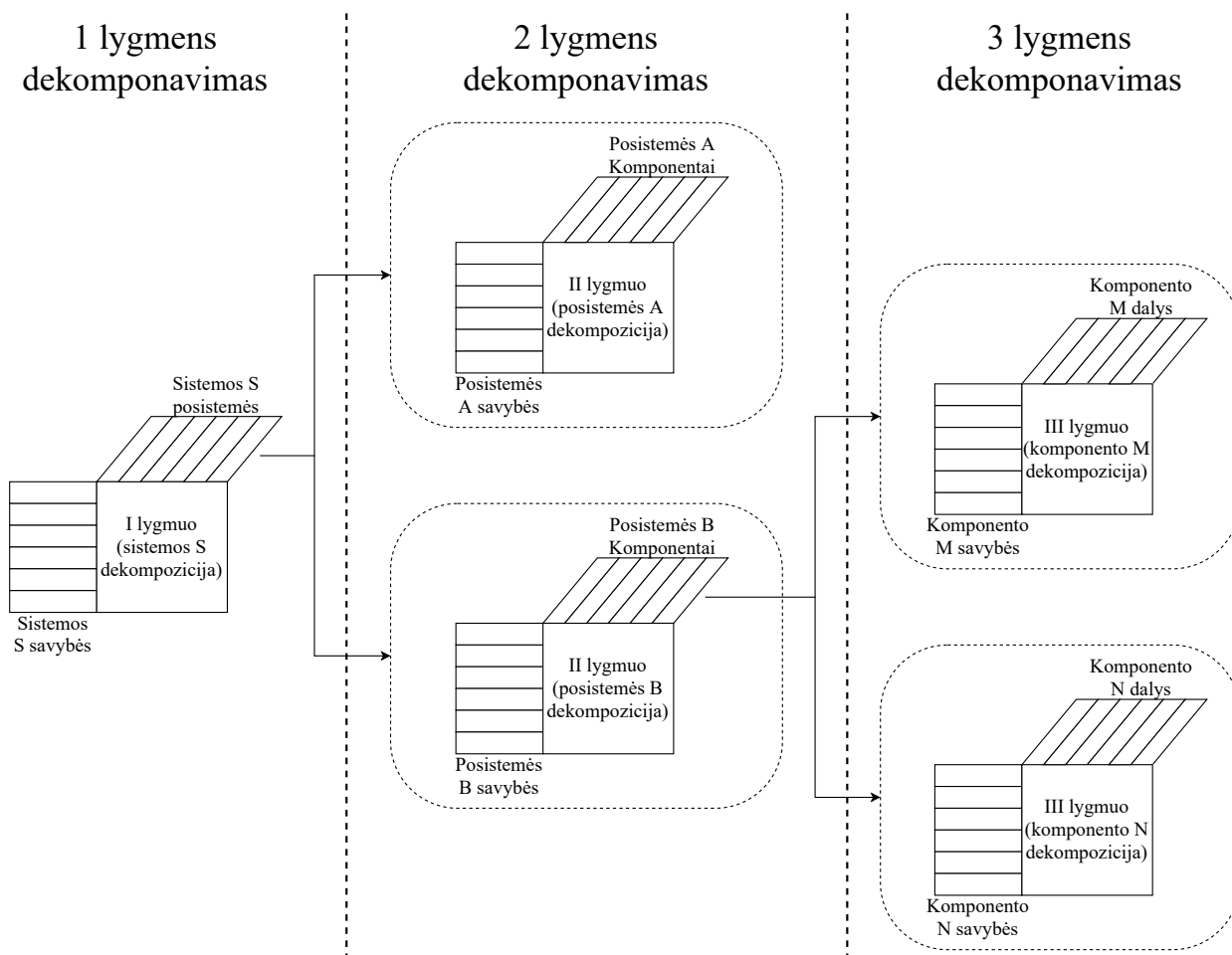
17 pav. KFS kortų perkėlimo į žemesnį lygmenį principas

Duotajame pavyzdyje yra pasirinkta IT sistema. Aukščiausiame lygmenyje yra susiejami kliento reikalavimai su projektavimo (techniniais) reikalavimais. Tuomet projektavimo reikalavimai perkeliama į žemesnį lygmenį, kuriame jie yra susiejami su programavimo reikalavimais. Pastarieji atitinkamai yra perkeliama į žemesnį lygmenį ir susiejami su testavimo reikalavimais.

3.2. Kortų pritaikymas dekomponavimo procesui

Originali KFS metodika nėra skirta dekomponavimui atlikti. Tačiau galima sukurti atskirą metodą, kuris pritaikytų esminius KFS kortų pildymo ir perkėlimo į žemesnius lygmenis principus. Toliau bus pristatomas dekomponavimo kortų metodas, kurį parengė šio darbo autorius. Tiriomojo darbo metu iškeltos problemos sprendimo įgyvendinimui sudarytas DK metodas savo esme grindžiamas KFS metodika.

Tačiau net ir pakartotinai panaudojus tam tikras KFS metodikos dalis, dekomponavimui pritaikytas metodas smarkiai skirsis. Tuo įsitikinti galima palyginus kortų perkėlimo į žemesnį lygį principo taikymą dekomponavimo (žr. 18 pav.) ir KFS (žr. 17 pav.) kortoms.



18 pav. Kortų perkėlimo į žemesnį lygmenį principo pritaikymas dekomponavimo procesui

Sakykime, jog turime sistemą S. Šią sistemą sudaro posistemės A ir B, o posistemė B yra sudaryta iš komponentų M ir N. Taigi, minėtoje sistemoje galime identifikuoti 5 elementus: pačią sistemą S, posistemas A ir B bei komponentus M ir N. Kiekvienas iš penkių įvardintų elementų turi šiuos individualius atributus: elemento savybes ir elemento sudedamąsias dalis. Taip pat galime identifikuoti 3 skirtingus abstrakcijų lygmenis: sistemos lygmuo, posistemių lygmuo ir komponentų lygmuo (atitinkamai 1, 2 ir 3 lygmenys). Galiausiai, kiekviena elemento sudedamoji dalis yra traktuojama kaip atskiras elementas žemesniame lygmenyje. Pavyzdžiui, 2 lygmenyje esantis posistemės B elementas yra sudarytas iš sudedamųjų dalių M ir N, todėl 3 lygmenyje komponentai M ir N bus traktuojami kaip atskiri elementai.

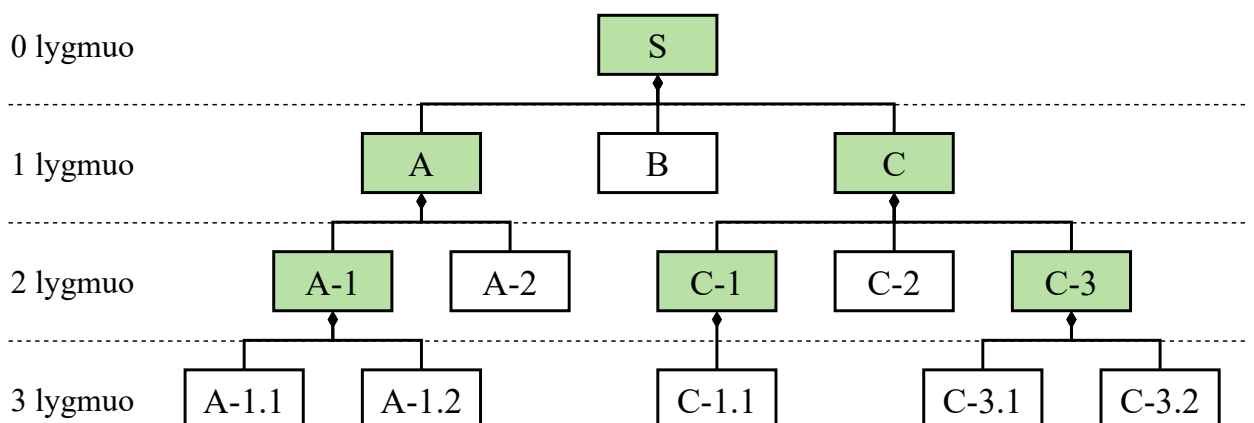
18 pav. iliustruoja aprašytos sistemos pavyzdį tariant, jog dekompozicija atliekama naudojant kortas. Iš iliustracijos seka du esminiai DK metodo principai:

1. Kiekvienas sistemos elementas, įskaitant pačią sistemą, turi savo unikalių savybių rinkinį bei sudedamųjų dalių, į kurias yra dekomponuojama, sąrašą.
2. Kiekviena elemento sudedamoji dalis egzistuoja kaip atskiras elementas žemesniame abstrakcijos lygmenyje (išskyrus neskaidomas sudedamąsias dalis).

Remiantis šiais dekomponavimo principais galime pabrėžti keletą skirtumų tarp KFS kokybės kortų metodo ir DK metodo:

- *Kortų kiekio augimo tempas.* Iš 17 pav. pateikto kortų eiliškumo galima išvelgti tiesinį augimą, t. y. kiekviename lygmenyje atsiranda viena ir tik viena korta. Tuo tarpu 18 pav. kortų augimas priklauso nuo to, į kiek sudedamųjų dalių duotame lygmenyje yra dekomponuojamas elementas. DK metodo atveju kortų kiekio augimas gali tapti labai spartus.
- *Kortos prasmė.* KFS atveju kiekviena korta atitinka atskirą reikalavimų lygmenį, pavyzdžiui kliento reikalavimų lygmenį. Tačiau DK metodo kiekviena korta atitinka atskiro elemento dekompoziciją (pavyzdžiui, antrame lygmenyje esančio B elemento dekompoziciją).
- *Kortos dalių atsikartojimas žemesniuose lygmenyse.* KFS atveju skirtumas tarp 1 ir 2 lygmens yra toks, kad visi pirmame lygmenyje esantys viršutinės dalies reikalavimai antrame lygmenyje atsideria kairiojoje dalyje. Tačiau dekomponavimo kortos prasmė yra konkretaus elemento dekompozicija, todėl kortos dalys nepersikelia į žemesnius lygmenis ir visur išlieka tokie patys: kairėje – elemento savybės, o viršuje – sudedamosios dalys.

Antruoju DK metodo principu teigiama, kad kiekviena elemento sudedamoji dalis egzistuoja kaip atskiras elementas žemesniame lygmenyje. Tačiau yra viena išimtis. Nors 18 pav. pavaizduotas išsišakojimas yra horizontalus, jis yra panašus į vertikalią sistemos dekomponavimo struktūrą, pateiktą 3 pav. Jeigu 3 pav. pateiktai sistemai pritaikytume 18 pav. esančias dekomponavimo kortas, gautume tokį kortų išsidėstymą, kuris pateiktas 19 pav.



19 pav. Dekomponavimo kortų išdėstymas dekompozicijos medyje

Kiekvienas paryškinta spalva pavaizduotas elementas reiškia tai, jog jam yra reikalinga dekomponavimo korta. Savo kortų neturi tik dekomponavimo medžio lapai, t. y. tie elementai, kurie toliau nėra dekomponuojami⁵.

Svarbu atkreipti dėmesį į tai, kad DK metodą galima taikyti tik hierarchija grindžiamiems dekomponavimo būdams. Šio darbo apimtyje nebuvo nagrinėjama arba rasta tokių dekomponavimo būdų, kurie būtų dekomponuojami ne hierarchiniu būdu. Tačiau neatmetama galimybė, kad tokie dekomponavimo būdai gali egzistuoti ir jiems šis metodas netiktų.

3.3. Informacinių sistemų dekomponavimo kortų metodas

Šiame poskyryje supažindinama su DK metodo turiniu: apibūdinamas kiekvienas kortos atributas, pateikti paaiškinimai, kaip gauti jų reikšmes, aptariama atributų prasmė dekomponavimo procesui ir galimos tendencijos, kurių atpažinimas gali padėti pasirinkti tinkamesnes alternatyvas. Tačiau prieš tai yra svarbu įvardinti, kokia yra DK metodo prasmė ir tikslas.

Dekomponuojant sistemas yra remiamasi architektų kompetencijomis ir turimomis žiniomis apie sistemą. Tačiau atlikti dekomponavimo veiklas nėra trivialu. Net ir turint patirties yra sudėtinga parinkti architektūrinius sprendinius. Be to, su architektūrinių sprendinių parinkimu architektas turi užtikrinti, kad kuriama sistema bus tokia, kokios ir reikia, t. y. bus validi.

Taip pat gali būti pamirštama arba sąmoningai pasirenkama neplėtoti alternatyvų, o ir esant alternatyvoms yra sunku lyginti jas tarpusavyje ir išlaikyti joms deramą dėmesį. Trūksta priemonių interaktyviai stebėti kiekvieno sprendimo poveikį ir nuspręsti, kuris dekomponavimo būdas yra labiau tinkamas pagal numatytus reikalavimus ir mažinti sistemos sudėtingumą.

Taigi, siekiant išspręsti minėtąsias problemas, DK metodo pagrindiniai tikslai yra šie:

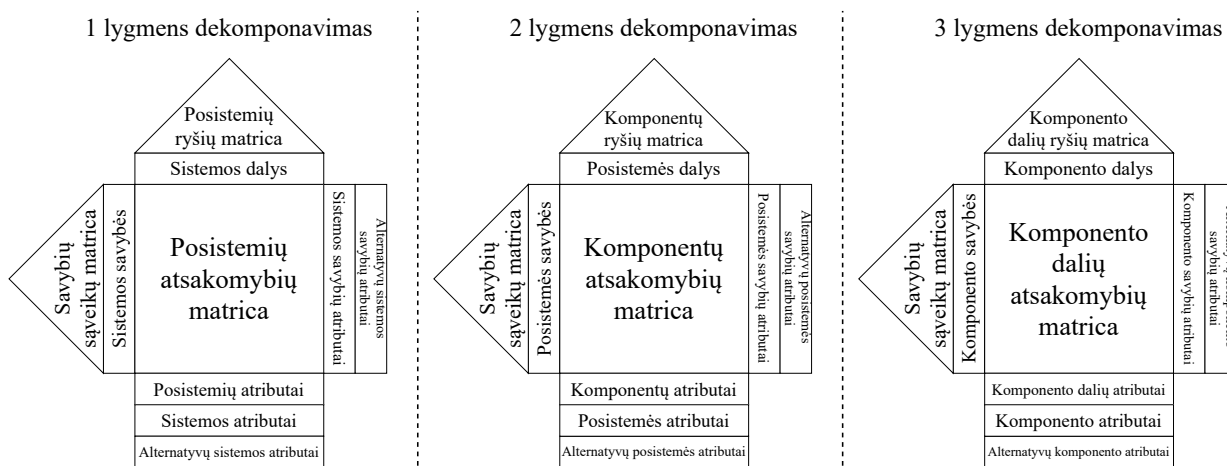
1. transformuoti sistemos reikalavimus į sistemos architektūrinius sprendinius;
2. užtikrinti, kad vykdant dekompoziciją nebūtų ignoruoti ar praleisti jokie reikalavimai;
3. įvertinti ir palyginti skirtingas dekompozicijos alternatyvas ir pasirinkti labiau tinkamą.

Šiuo metodu papildomai siekiama pateikti sistemos ir jos dalių informaciją lengvai suprantamu būdu, suteikti galimybę stebėti, kaip nuo konkrečių sprendimų keičiasi kiekybinių atributų reikšmės. Taip pat padidinti ir užtikrinti sklandesnę sistemų architekto tarpusavio supratimą ir susikalbėjimą su reikalavimų inžinierių komanda.

⁵Dekompozicijos medžio lapams nėra griežtai draudžiama turėti savo kortas. Tačiau jas pildyti nėra prasminga, nes būtų užpildoma tik eilučių informacija – atominių neskaidomų elementų savybės. Tuo tarpu kortos dalys, susijusios su sudedamosiomis dalimis liktų tuščios. Autoriaus nuomone, atominių elementų savybių dokumentavimui galima pasirinkti tam labiau pritaikytus metodus.

3.3.1. Dekomponavimo kortos turinys

Kiekvienas dekomponavimo lygmuo pasižymi savo ypatumais. Nagrinėjant 3.2 poskyryje pateiktą sistemos pavyzdį, pirmame lygmenyje naudojamos sistemos ir posistemių sąvokos, antrame lygmenyje – posistemių ir komponentų sąvokos, trečiame lygmenyje – komponentų ir komponentų dalių sąvokos. Tokiu atveju dekomponavimo kortos skirtinguose lygmenyse atrodo taip, kaip pavaizduota 20 pav.

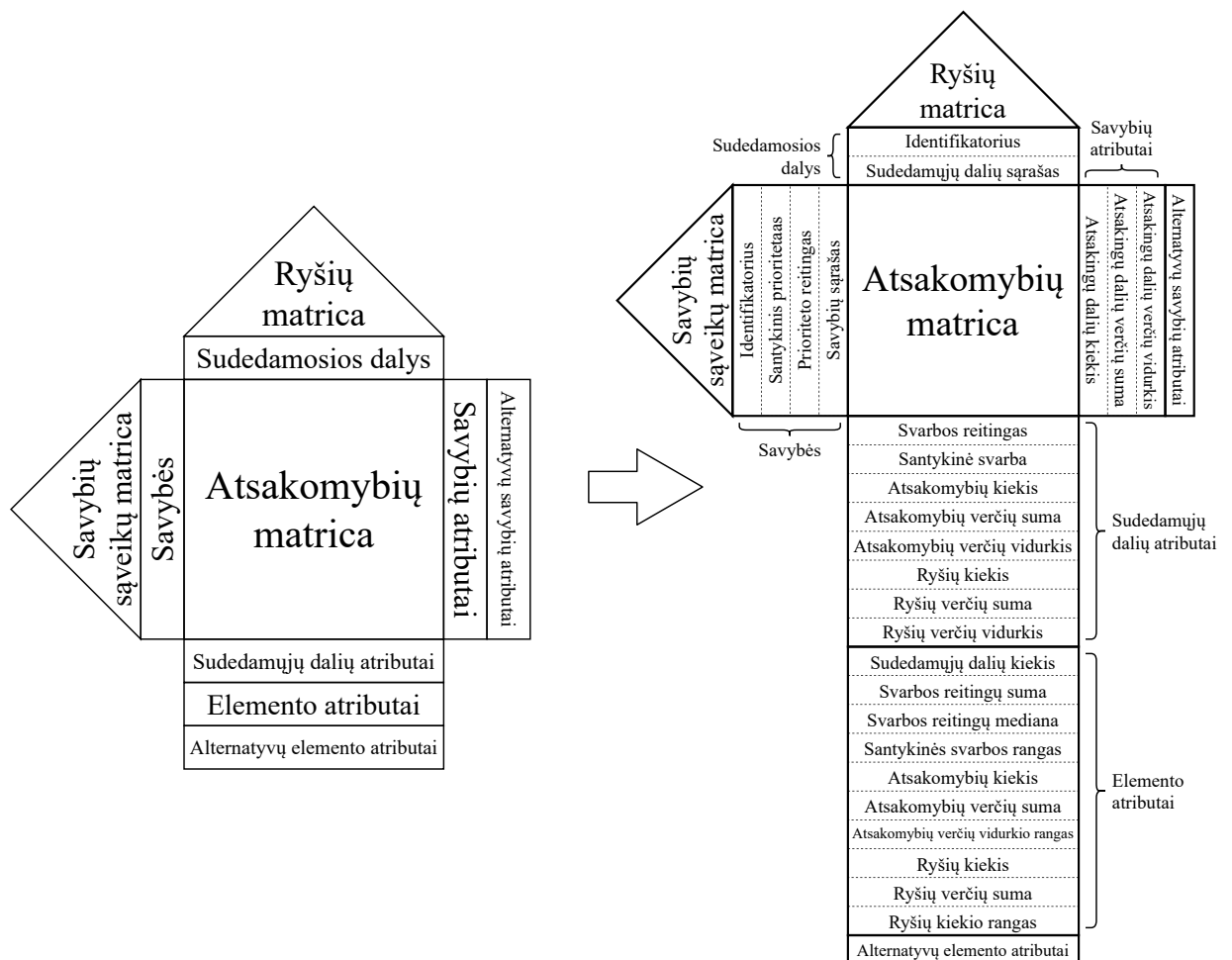


20 pav. Dekomponavimo kortų turinys skirtinguose lygmenyse

Dėl terminų gausos gali pasunkėti kortų dalių atpažinimas. Todėl yra prasmingas kortos apibrėžimas, kuris tinka bendru atveju. Tokia dekomponavimo korta, tinkanti bet kuriame abstrakcijos lygmenyje, pateikta 21 pav. kairėje pusėje. Korta yra universali, nes nėra prisirišama prie konkretaus abstrakcijos lygmens terminijos.

Be to, korta, esanti kairėje paveikslėlio pusėje, yra suskleista, t. y. kortos dalys: „Savybės“, „Sudedamosios dalys“, „Sudedamųjų dalių atributai“, „Savybių atributai“, „Elemento atributai“ yra atributų grupės, apimančios keletą atributų. Dešinėje pusėje esanti korta (žr. 21 pav.) yra pilnai išskleista, kurioje išdėstyti visi kortoje esantys atributai.

Tolesniuose skirsniuose bus nagrinėjami visi kortoje esantys atributai. Norint geriau suprasti pateiktą medžiagą, autorius rekomenduoja susipažinti su vartojama terminija, kuri pateikta šio darbo sąvokų skyriuje (žr. 83 psl.)



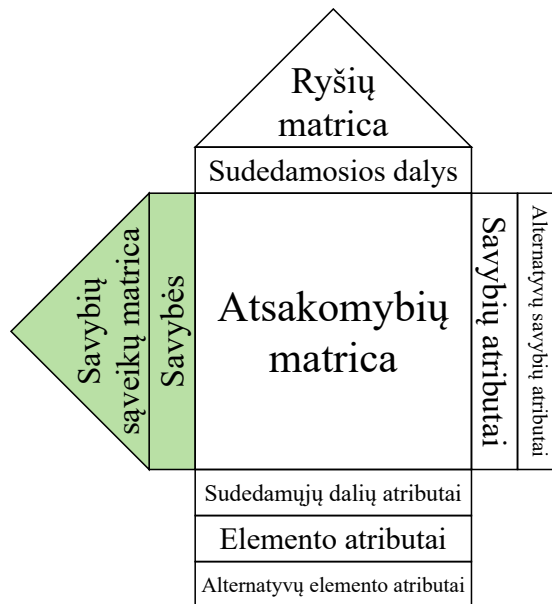
21 pav. Suskleista ir išskleista dekomponavimo korta bendruoju atveju

Dekomponavimo kortas būtina pildyti korektiškai tokiu eiliškumu, kuris yra pateiktas tolimesniuose poskyriuose ir laikantis nurodytų taisyklių. Šio darbo apimtyje DK metodo pagrindu buvo sukurtas specialus dekomponavimui skirtas įrankis, kurį galima atsisiųsti iš viešos darbo autoriaus „gitHub“ repozitorijos⁶. Priemonę buvo pasirinkta sukurti „Microsoft Excel“ aplinkoje. Dekomponavimo kortų metaduomenys, apimantys atributų pildymo eiliškumą bei kitas taisykles parengto įrankio kontekste yra pateiktos priede nr. 1.

3.3.2. Kairioji dekomponavimo kortos dalis

22 pav. paryškintos savybių atributų grupės ir savybių sąveikų matricos kortos dalys, išdėstytos kairėje dekomponavimo kortos pusėje.

⁶Žr. <https://github.com/gedgeds/Decomposition-Cards-Method>; failas – „Dekomponavimo kortų įrankis.xlsx“; tikrinta 2021-05-09.



22 pav. Kairioji dekomponavimo kortos dalis

3.3.2.1. Savybių sąrašas

Savybių sąrašo dalyje į eilutes yra surašomos siekiamos nagrinėjamo elemento savybės. Jos turėtų būti parenkamos iš 2.6 poskyryje pateikto dekomponavimo reikalavimų katalogo. Šią dalį turėtų užpildyti reikalavimų inžinierius, atlikęs reikalavimų išsiaiškinimo darbą, o dekomponavimo kortos pildymas turėtų prasidėti nuo šio atributo pildymo. Kartu su savybių sąrašu toliau yra pildomi atributai kiekvienai individualiai nagrinėjamo elemento savybei atskirai.

3.3.2.2. Identifikatorius

Tai automatinis savybės atributas. Identifikatorius leidžia identifikuoti ir vienareikšmiškai atskirti kiekvieną unikalią elemento savybę. Identifikatorių žymėjimą galima pasirinkti savo nuožiūra, gali būti naudojami specialūs identifikavimo formatai su raidiniais ir skaitiniais simboliais, tačiau rekomenduotina identifikuoti naudojant eilučių numerius, t. y. $J = \{1, 2, \dots, m\}$, kur J – savybių identifikatorių aibė, o m – nagrinėjamo elemento savybių kiekis.

3.3.2.3. Prioriteto reitingas

Tai bazinis kiekybinis savybės atributas. Prioriteto reitingas parodo, kokia yra kiekybinė savybės svarba nagrinėjamo elemento kontekste. Dėl ribotų išteklių visų siekiamų savybių nebūtinai pavyks užtikrinti, todėl yra prasminga joms priskirti prioritetus. Tokiu būdu savybės yra kategorizuojamos į būtinąsias bei mažiau svarbias.

Prieš pradėdant pildyti prioritetų reitingus būtina apibrėžti, kokioje skalėje bus vertinami prioritetai. Kad kuo didesnė skalė, tuo didesnis bus prioritetų tikslumas. Pavyzdžiui, skalė nuo 1 iki 10 užtikrina didesnę tikslumą, negu skalė nuo 1 iki 5, su kuria dalis informacijos gali būti prarandama. Atliekant vertinimą reikia turėti omenyje, kad, kuo aukštesnis reitingo įvertis, tuo didesnis savybės prioritetas.

DK metodas nepateikia nurodymų, kaip tiksliai įvertinti konkrečių savybių prioritetus – tai yra paliekama reikalavimų inžinieriaus kompetencijai. Tačiau konkrečias skaitines vertes gauti gali padėti savybių tarpusavio lyginimo matricos metodas, kurį taikant savybių svarba nustatoma lyginant kiekvieną tarpusavyje.

Prioritetų priskyrimas gali turėti įtakos tolimesniam dekomponavimo procesui. Pavyzdžiui, atsižvelgdamas į prioritetus sistemos architektas gali dekomponuoti elementus į dalis taip, kad svarbiausias savybes realizuotų konkreti, būtent jai skirta sudedamoji elemento dalis. Arba dekomponuoti išvengiant situacijų, kada sistemos dalis, įgyvendinanti labai svarbią savybę, kartu įgyvendina ir mažiau svarbią savybę, nes tai galimai sukurtų nepageidautinas savybių priklausomybes.

3.3.2.4. Santykinis prioriteto reitingas

Tai išvestinis kiekybinis atributas, nusakantis konkrečios savybės svarbą visų nagrinėjamo elemento savybių atžvilgiu.

Norint apskaičiuoti atributo reikšmę, visų pirma būtina užpildyti visų savybių prioritetų reitingus (žr. 3.3.2.3 skirsnį). Tada reikia taikyti formulę:

$$f(y) = \frac{P(y)}{\sum_{j=1}^m P(y_j)} \cdot 100\%, m > 0. \quad (1)$$

Čia y – savybė; m – visų nagrinėjamo elemento savybių kiekis; $P(y)$ – savybės y prioriteto reitingo reikšmė.

Pagrindinė tendencija, į kurią reikia atkreipti dėmesį, tai kai daugumos savybių santykinio prioriteto reitingo reikšmės sutampa arba beveik sutampa. Taip gali nutikti dėl kelių priežasčių. Pirma, jeigu yra pasirinkta per siaura prioriteto reitingo skalė. Tokiu atveju daugelis savybių įgauna identiškus prioritetus, nors jie turėtų skirtis. Norint išvengti šio scenarijaus, reikia padidinti prioritetų reitingų skalę ir iš naujo atlikti prioritetų vertinimą.

Antra, taip gali nutikti, jeigu reikalavimų inžinierius daugumą savybių vertina kaip svarbias, šitaip daugumos savybių prioritetų reitingus nustatydamas tomis pačiomis vertėmis. To galima išvengti nustatant kvotas prioritetų reitingams ir iš naujo atlikti prioritetų vertinimą. Pavyzdžiui,

esant skalei nuo 1 iki 5, tą pačią prioriteto vertę galėtų įgauti ne daugiau kaip 40% savybių.

Santykinis savybės prioriteto reitingo atributas gali padėti priimti sprendimus dėl įgyvendinamų savybių. Pavyzdžiui, gali būti pasirinkta įgyvendinti tik tas savybes, kurių santykinis prioriteto reitingas didesnis negu 5%, arba iš viso įgyvendinti tik tokį savybių rinkinį, kurio bendra santykinio prioriteto reitingo reikšmių suma neviršija 80%.

3.3.2.5. Sąveikų matrica

Tai matricos tipo atributas, kuriame yra užpildomos sąveikos tarp visų savybių porų. Gali susiklostyti taip, kad nagrinėjamame elemente kelios savybės, kurias įgyvendins ta pati sudedamoji dalis, bus neutralios, prieštaraus arba papildys vienos kitas. Iš viso išskiriami penki savybių sąryšių tipai, kurių kiekvienas turi individualų žymėjimą:

1. *Stipri neigiama sąveika.* Tai ryšys tarp dviejų savybių, kai tarpusavyje jos yra stipriai priklausomos ir vienos iš jų pilnas įgyvendinimas turėtų stiprų neigiamą poveikį kitos savybės užtikrinimui. Tokiu atveju savybių sąveikų matricoje abiejų savybių susikirtimo langelyje yra pažymimi simboliai „- -“.
2. *Silpna neigiama sąveika.* Tai ryšys tarp dviejų savybių, kai tarpusavyje jos yra silpnai priklausomos ir vienos iš jų pilnas įgyvendinimas turėtų silpną neigiamą poveikį kitos savybės užtikrinimui. Tokiu atveju savybių sąveikų matricoje abiejų savybių susikirtimo langelyje yra pažymimas simbolis „-“.
3. *Neutrali sąveika.* Tai sąryšis tarp dviejų savybių, kai tarpusavyje jos yra nepriklausomos ir viena kitai poveikio nedaro. Tokiu atveju savybių sąveikų matricoje abiejų savybių susikirtimo langelis yra paliekamas tuščias.
4. *Silpna teigiama sąveika.* Tai ryšys tarp dviejų savybių, kai tarpusavyje jos yra silpnai priklausomos ir vienos iš jų pilnas įgyvendinimas turėtų silpną teigiamą poveikį kitos savybės užtikrinimui. Tokiu atveju savybių sąveikų matricoje abiejų savybių susikirtimo langelyje yra pažymimas simbolis „+“.
5. *Stipri teigiama sąveika.* Tai ryšys tarp dviejų savybių, kai tarpusavyje jos yra stipriai priklausomos ir vienos iš jų pilnas įgyvendinimas turėtų stiprų teigiamą poveikį kitos savybės užtikrinimui. Tokiu atveju savybių sąveikų matricoje abiejų savybių susikirtimo langelyje yra pažymimi simboliai „+ +“.

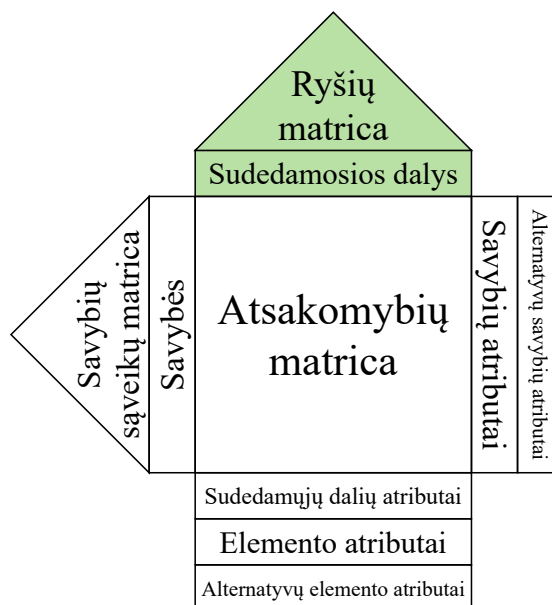
DK metodas nepateikia taisyklių, kuriomis remiantis reikėtų vertinti konkrečių savybių są-

veikas – tai yra paliekama reikalavimų inžinieriaus kompetencijai. Tačiau konkrečius vertinimus gauti gali padėti taikomas „i*“ metodas [DFH16], NFR karkasas [YM09] arba kitos pagalbinės priemonės.

Savybių sąveikų matrica yra naudinga ne tik reikalavimų inžinieriui, kad būtų kuo anksčiau identifikuoti neigiami sąryšiai, bet ir architektui. Architektas, dekomponuodamas sistemą gali siekti vengti situacijų, kai elemento sudedamoji dalis tampa atsakinga už kelių savybių, tarpusavyje turinčių neigiamas priklausomybes, įgyvendinimą. Tokiu atveju būtų dekomponuojama taip, kad už visų prieštaraujančių savybių įgyvendinimą būtų atsakingos skirtingos sudedamosios dalys.

3.3.3. Viršutinė dekomponavimo kortos dalis

23 pav. paryškintos sudedamųjų dalių atributų grupės ir tų dalių ryšių matricos kortos dalys, išdėstytos dekomponavimo kortos viršuje.



23 pav. Viršutinė dekomponavimo kortos dalis

3.3.3.1. Sudedamųjų dalių sąrašas

Sudedamųjų dalių sąrašo dalyje į stulpelius yra surašomas nagrinėjamo elemento dekomponavimo rezultatas – sudedamosios dalys. Šį atributą reikėtų užpildyti po to, kai baigiama pildyti 3.3.2 pateiktos dalys. Šį elementą turėtų užpildyti architektas, įvertinęs 3.3.2 dalyje užpildytus duomenis ir pasirinkęs dekomponavimo būdą. DK metodas nepateikia konkrečių būdų į kokias dalis dekomponuoti nagrinėjamą elementą – tai yra paliekama architekto kompetencijai. Tačiau būtina laikytis tam tikrų principų:

1. Dekompozicija reikalinga sistemos bei jos dalių sudėtingumo mažinimui. Tai reiškia, jog architektas turi remtis turimais duomenimis apie sistemą ar jos dalis bei lyginti jos sudėtingumui daromą dekomponavimo sprendimų ir alternatyvų įtaką.
2. Dekompozicijos rezultatas turi užtikrinti siekiamų savybių realizaciją. Priešingu atveju bus sukurta nevalidi sistema. Tai nebūtinai reiškia, kad privalo būti įgyvendintos visos savybės arba su reikalavimų inžinieriumi suderintas apsibrėžtų savybių rinkinys (3.3.2.4 skirsnyje pateiktas scenarijus, kada įgyvendinama 80% savybių).
3. Reikia laikytis nuoseklumo pasirinkto dekomponavimo būdo atžvilgiu. Tai reiškia, jog dekomponavimas neturėtų būti chaotiškas. Kiekvienas dekomponavimo būdas turėtų turėti kriterijus, kurių reikėtų laikytis. Dekomponavimo būdų literatūros analizė pateikta 2.5 poskyryje.

3.3.3.2. Identifikatorius

Identifikatoriaus reikšmės kiekvienai sudedamajai elemento daliai pildomos remiantis tokiu pat būdu, kuris pateiktas 3.3.2.2 skirsnyje. Sudedamąsias dalis rekomenduotina identifikuoti naudojant eilučių numerius, t. y. $I = \{1, 2, \dots, n\}$, kur I – savybių identifikatorių aibė, o n – nagrinėjamo elemento siekiamų savybių kiekis.

3.3.3.3. Ryšių matrica

Tai matricos tipo atributas. Kiekvienas sudedamųjų dalių sąrašo elementas turi priklausomybes – ryšius su visomis kitomis sudedamosiomis dalimis. Šių ryšių pobūdis yra užpildomas ryšių matricoje. Ryšiai gali būti trijų skirtingų tipų, kurių kiekvienas matricoje turi žymėjimo simbolius bei kiekybinę ryšio reikšmę:

1. *Neegzistuojantis ryšys*. Tai ryšys tarp dviejų dalių, kai dalys tarpusavyje yra visiškai nesusijusios ir viena apie kitą nežino. Tokiu atveju ryšių matricoje dalių susikirtimo langelis yra paliekamas tuščias ir kiekybinė reikšmė yra 0.
2. *Silpnas ryšys*. Tai ryšys tarp dviejų dalių, kai tarp dalių yra vienpusis arba abipusis informacijos apsikeitimas, tačiau šios transakcijos neturi arba turi silpną poveikį dalims ir be šių transakcijų dalys galėtų funkcionuoti. Tokiu atveju ryšių matricoje dalių susikirtimo langelyje yra pažymimas simboliu „o“ ir kiekybinė reikšmė yra 1.
3. *Stiprus ryšys*. Tai ryšys tarp dviejų dalių, kai tarp dalių yra vienpusis arba abipusis informacijos apsikeitimas. Šios transakcijos turi stiprų poveikį dalims ir be šių transakcijų viena arba

abi dalys negalėtų funkcionuoti arba jų egzistavimas būtų neprasmingas. Tokiu atveju ryšių matricoje dalių susikirtimo langelyje yra atliekamas simbolio „●“ žymėjimas ir kiekybinė reikšmė yra 3.

Nors matricoje yra pateikiami tiesioginiai sudedamųjų dalių sąryšiai, esant poreikiui, galima atsekti ir netiesioginius sąryšius tarp 2 elementų, net jeigu tarp jų tiesioginis ryšys ir neegzistuoja. Šis pavyzdys iliustruotas 24 pav.

Dalis X	Dalis Y	Kiekybinė reikšmė
a	c	0
a	b	2
b	c	2

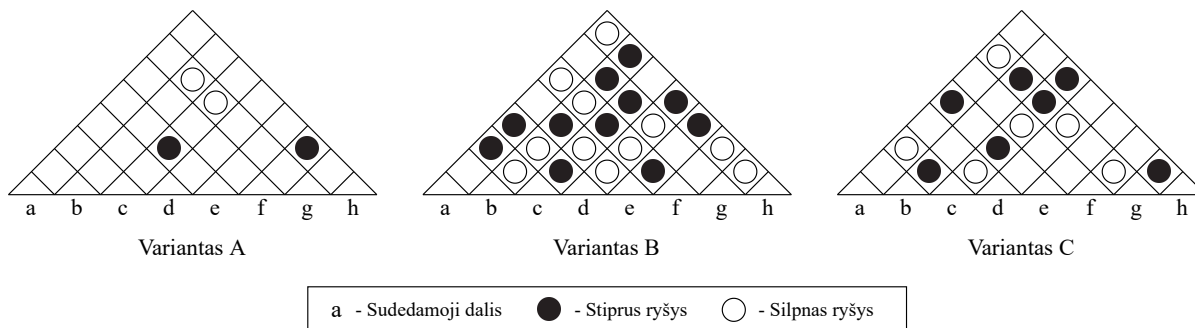
24 pav. Netiesioginio ryšio radimas

Čia dalis a neturi tiesioginio ryšio su dalimi c. Tačiau dalis a turi stiprų ryšį su dalimi b, o dalis b – stiprų ryšį su dalimi c. Taigi, dalis a turi netiesioginį sąryšį su dalimi c.

Sudedamųjų dalių ryšių matricoje galima atpažinti tam tikras tendencijas, pavaizduotas 25 pav. variantuose:

- *Dauguma matricos langelių yra tušti* (žr. 25 pav. A variantas). Tai reiškia labai mažą nagrinėjamo elemento dalių sukibimą. Toks variantas gali pasireikšti dekomponuojant dalis į savarankiškus vienetus, kurie problemas sprendžia lokaliai, todėl jiems nereikia bendrauti su išorinėmis dalimis. Tiesa, nedaug ryšių tarp sudedamųjų dalių gali susidaryti ir kitomis aplinkybėmis. Norint sumažinti sudėtingumą, šis variantas yra siektinas.
- *Dauguma matricos langelių turi „o“ arba „●“ ryšius* (žr. 25 pav. B variantas). Tokiu atveju nagrinėjamo elemento dalys bus tankiai sukibusios ir tai gerokai padidins sudėtingumą. Toks variantas gali pasireikšti kai nagrinėjamas elementas turi daug atsakomybių. Esant šiam scenarijui, tinkamas sprendimas būtų grįžti į aukštesnį lygį ir sumažinti sudedamosios dalies, atitinkančios nagrinėjamą elementą, atsakomybių kiekį. B variantas taip pat gali pasireikšti dekomponuojant elementą į tokias dalis, kurios turi nemažai atsakomybių. Tokiu atveju reikėtų dekomponuoti iš naujo, priskiriant kuo mažiau atsakomybių kiekvienai sudedamajai daliai (jeigu tai įmanoma).
- *Tam tikros sudedamosios dalys turi daug ryšių* (žr. 25 pav. C variantas, c ir g dalys). Šios dvi dalys yra glaudžiai susijusios su didžiaja dalimi kitų dalių. Tai gali indikuoti apie koordinuo-

jančias arba pagalbines dalis, kurių egzistavimas ne visada išvengiamas. Tačiau tokių dalių egzistavimas taip pat gali signalizuoti ir apie nekokybišką dekompoziciją, kai dekomponuojama į milžiniškas dalis, atsakingas už didžiąją dalį nagrinėjamo elemento savybių. Vienas iš būdų to išvengti – suskaidyti milžinišką dalį esamajame lygmenyje į smulkesnes dalis.

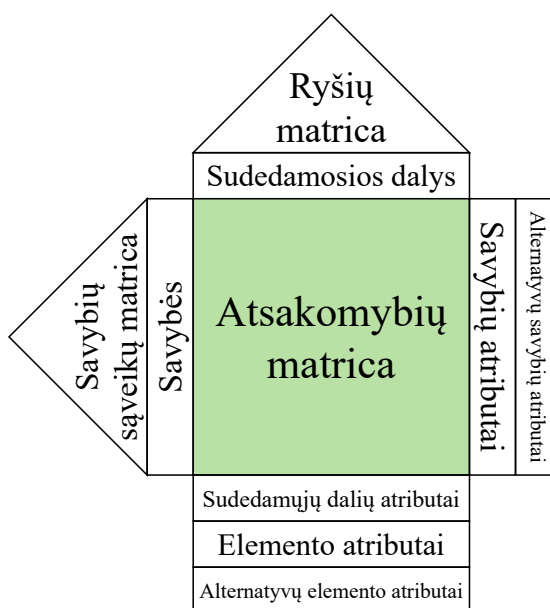


25 pav. Ryšių matricos atributo tendencijų variantai

Ši matrica yra vienas iš svarbiausių dekomponavimo kortos atributų. Ji padeda nustatyti ryšius tarp sudedamųjų elemento dalių, šitaip prisidedama prie sudėtingumo vertinimo taikant konkretų dekomponavimo būdą. Dekomponuojant elementą skirtingais būdais, skirsis ir sąryšiai tarp jų, todėl galima ieškoti alternatyvių būdų dekomponuoti elementus taip, kad matricoje būtų kuo daugiau tuščių langelių.

3.3.4. Centrinė dekomponavimo kortos dalis

26 pav. paryškintas atsakomybių matricos atributas, esantis centrinėje dekomponavimo kortos dalyje.



26 pav. Centrinė dekomponavimo kortos dalis

3.3.4.1. Atsakomybių matrica

Tai matricos atributas, skirtas dokumentuoti 3.3.2.1 ir 3.3.3.1 skirsniuose aprašytų elemento savybių ir dekomponavimo metu gautų sudedamųjų dalių atributų tarpusavio ryšius.

Architekto atsakomybė yra dekomponuoti sistemą ar jos elementus taip, kad būtų užtikrintas reikalavimų inžinieriaus pateiktas savybių sąrašas. Šių savybių įgyvendinimą gali užtikrinti tik sistemos sudedamosios dalys. Įsitikinti, kad savybės tikrai bus įgyvendintos, galima priskiriant konkrečias atsakomybes sistemos dalims, šitaip užtikrinant atsekamumą. Dėl to atsakomybių matrica yra vienas svarbiausių dekomponavimo kortos atributų.

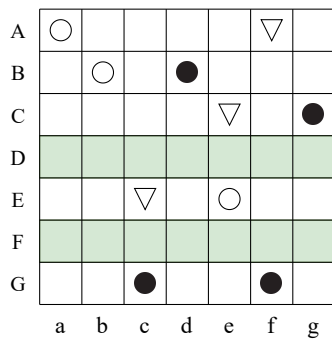
Ryšiai matricoje gali būti keturių skirtingų tipų, kurių kiekvienas turi žymėjimo simbolius bei atsakomybės kiekybinę reikšmę:

1. *Neegzistuojanti atsakomybė*. Atsakomybė yra neegzistuojanti, kada sudedamoji dalis visiškai neprisideda prie savybės užtikrinimo. Tokiu atveju matricoje sudedamosios dalies ir savybės susikirtimo langelis yra paliekamas tuščias, o atsakomybės reikšmė yra 0.
2. *Silpna (pagalbinė) atsakomybė*. Atsakomybė yra silpna, kai sudedamoji dalis nežymiai arba netiesiogiai prisideda prie savybės užtikrinimo. Pavyzdžiui, jeigu sudedamoji dalis netiesioginiu būdu prisideda prie savybės užtikrinimo, pagelbėdama kitoms sudedamosioms dalims realizuoti savybę. Taip pat ir tuo atveju, jeigu sudedamoji dalis tiesiogiai prisideda prie savybės užtikrinimo, tačiau neįgyvendinus sudedamosios dalies, savybės užtikrinimas nagrinėjamame elemente nuo to nenukentėtų. Tokiu atveju matricoje sudedamosios dalies ir savybės susikirtimo langelyje yra pažymimas simbolis „ ∇ “, o atsakomybės reikšmė yra 1.
3. *Vidutinė atsakomybė*. Atsakomybė yra vidutinė, kai sudedamoji dalis svariai prisideda prie savybės užtikrinimo. Pavyzdžiui, jeigu sudedamoji dalis būtų neįgyvendinta, savybės užtikrinimas nagrinėjamame elemente gerokai nukentėtų. Tokiu atveju matricoje sudedamosios dalies ir savybės susikirtimo langelyje yra pažymimas simbolis „ \circ “, o atsakomybės reikšmė yra 3.
4. *Stipri (kritinė) atsakomybė*. Atsakomybė yra stipri, kada sudedamoji dalis kritiškai prisideda prie savybės užtikrinimo. Pavyzdžiui, jeigu neįgyvendinus šios sudedamosios dalies, savybė nagrinėjamame elemente būtų neužtikrinta. Tokiu atveju matricoje sudedamosios dalies ir savybės susikirtimo langelyje yra pažymimas simbolis „ \bullet “, o atsakomybės reikšmė yra 9.

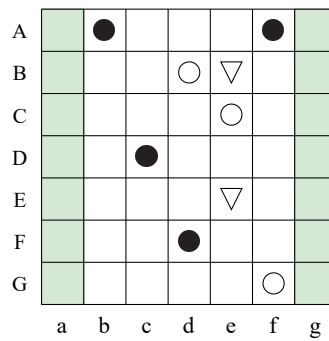
Šioje matricoje taip pat galima atpažinti kelias tendencijas, pateiktas 27 pav. Paryškintomis juostomis yra pažymėtos tos eilutės arba stulpeliai, kuriuose pasireiškia atitinkamos tendencijos:

- *Tuščios eilutės* (žr. 27 pav. 1 variantas, D ir F eilutės). Tai pasireiškia, kada nagrinėjamame elemente yra neįgyvendintų savybių. Tai reiškia, jog dekomponavimas atliktas ne iki galo sėkmingai, kadangi su turimu sudedamųjų dalių rinkiniu ne visos savybės yra realizuojamos. Taigi, dekompozicija yra nevalidi. Esant tokiam scenarijui būtina peržiūrėti dekompozicijos rezultatą ir pasirūpinti, kad sudedamųjų dalių rinkinys padengtų visas reikiamas savybes.
- *Tušti stulpeliai* (žr. 27 pav. 2 variantas, a ir g stulpeliai). Tai indikuoja apie perteklines sudedamąsias dalis. Kiekvienos papildomos dalies egzistavimas prisideda prie sistemos sudėtingumo, todėl, esant tokiam scenarijui, visas sudedamąsias dalis, kurios nėra atsakingos nei už vieną savybę, yra prasminga pašalinti.
- *Eilutės be stiprių ryšių* (žr. 27 pav. 3 variantas, C ir G eilutės). Tendencija gali indikuoti, jog atitinkama savybė bus realizuota ne iki galo arba įgyvendinta paviršutiniškai. Idealiu atveju kiekviena savybė turėtų turėti stiprų ryšį su bent viena sudedamąja dalimi.
- *Stulpeliai su silpnomis atsakomybėmis* (žr. 27 pav. 4 variantas, b ir c stulpeliai). Tendencija indikuoja, kad atitinkamos sudedamosios dalys galimai yra pagalbinės. Nors tai nėra neigiama tendencija, ši dalis turės ryšius su daugeliu kitų dalių šitaip padidindama sistemos sudėtingumą. Dėl šios priežasties nagrinėjamame elemente vertėtų riboti tokių dalių kiekį.
- *Identiškos eilutės* (žr. 27 pav. 5 variantas, B, D ir E eilutės). Tokiu atveju pažymėtos savybės turėtų atsidurti žemesniame lygmenyje, o esamajame lygmenyje – apjungtos į visas šias savybes apgaubiantį vieną bendrą reikalavimą. Scenarijus gali pasireikšti, jeigu reikalavimų inžinierius elementui priskiria žemesnio lygmens reikalavimus, kurie turėtų būti priskirti elemento sudedamosioms dalims.
- *Identiški stulpeliai* (žr. 27 pav. 6 variantas, d ir g stulpeliai). Kadangi sudedamosios dalys turi vienodas atsakomybes, tikėtina, jog nagrinėjamame elemente jos išskaidytos per smulkiai. Tai nėra gerai, nes kuo daugiau sudedamųjų dalių, tuo labiau išauga sudėtingumas. Tam, kad to būtų išvengta, rekomenduotina atitinkamas dalis apjungti į vieną stambesnę dalį, kuri apimtų visų šių smulkesnių dalių atsakomybes.
- *Perpildytos eilutės* (žr. 27 pav. 7 variantas, C ir F eilutės). Ši tendencija gali indikuoti, jog savybė yra nelokalizuojama. Taigi, savybė turėtų būti iškelta į aukštesnį lygmenį, kuriame už savybę būtų atsakingas nagrinėjamas elementas, o esamajame lygmenyje savybę reikėtų suskaidyti į smulkesnius reikalavimus.
- *Perpildyti stulpeliai* (žr. 27 pav. 8 variantas, b ir d stulpeliai). Tendencija gali indikuoti apie milžiniškas sudedamąsias elemento dalis, turinčias per daug atsakomybių. Tokiu atveju esamajame lygmenyje yra prasminga dekomponuoti šią dalį į smulkesnes dalis ir perskirstyti

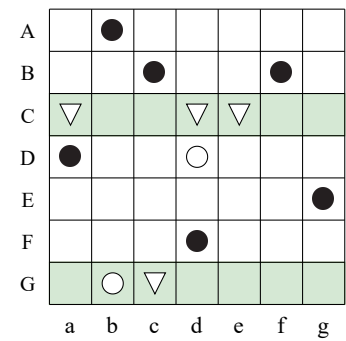
atsakomybes pastarosioms.



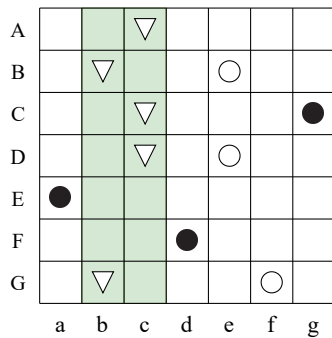
1 Variantas



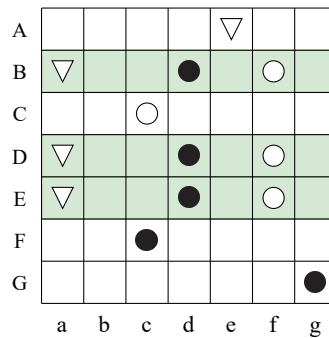
2 Variantas



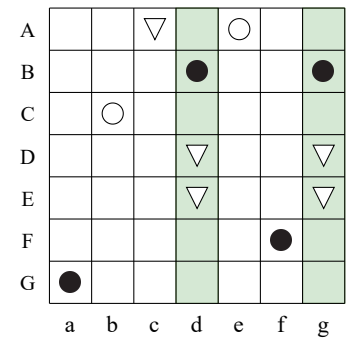
3 Variantas



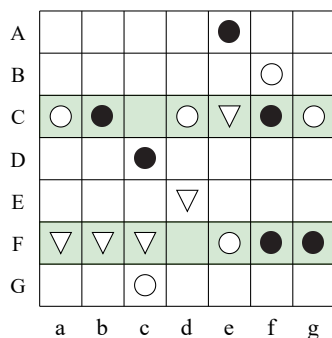
4 Variantas



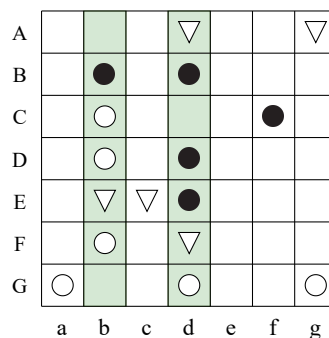
5 Variantas



6 Variantas



7 Variantas



8 Variantas

Legenda:

- ▽ - Silpna atsakomybė
- - Vidutinė atsakomybė
- - Stipri atsakomybė
- A - Savybė
- a - Dekomponuotas elementas

27 pav. Atsakomybių matricos atributo tendencijų variantai

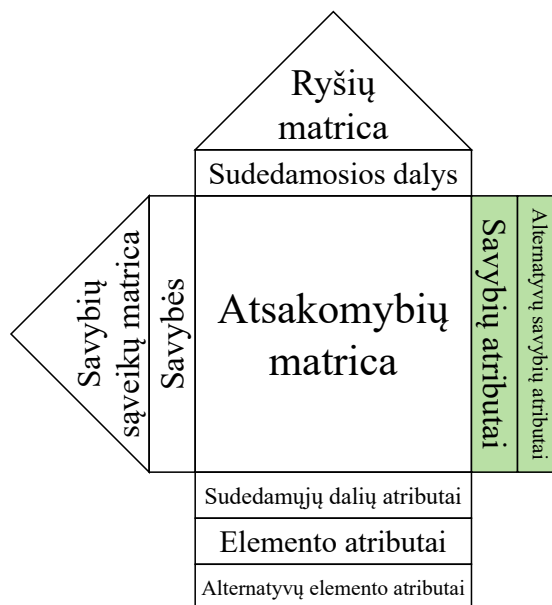
Identiškos eilutės bei identiški stulpeliai yra vienintelės tendencijos, kurias galima identifikuoti tik iš pačios matricos. Visas kitas tendencijas padeda identifikuoti atributai, pristatyti 3.3.5 ir 3.3.6 skirsniuose.

Kaip jau minėta, matrica garantuoja savybių ir sudedamųjų dalių ryšių atsekamumą, tad jos dėka palengvinamas dekomponavimo procesas. Atsakomybių matrica gali padėti identifikuoti tam tikras tendencijas, dekomponavimo trūkumus. Architektas, turėdamas šiuos duomenis, gali atlikti dekompozicijos korekcijas. Matrica taip pat leidžia vizualiai matyti apibendrintą dekomponavimo būdo rezultata, o jos kiekybinius duomenis naudoja dauguma kitų išvestinių atributų.

3.3.5. Dešinioji dekomponavimo kortos dalis

28 pav. paryškintos savybių atributų grupės ir alternatyvų savybių atributų grupės kortos dalys, išdėstytos dešinėje dekomponavimo kortos pusėje.

Visų pirma, dešinioji dekomponavimo kortos dalis pradedama pildyti nuo savybių atributų. Tai rinkinys išvestinių kiekybinių atributų, skirtų suteikti daugiau informacijos apie savybių būseną nagrinėjamame elemente naudojant pasirinktą dekomponavimo būdą. Atributų reikšmės pildomos kiekvienai individualiai savybei atskirai.



28 pav. Dešinioji dekomponavimo kortos dalis

3.3.5.1. Atsakingų dalių kiekis

Tai išvestinis kiekybinis savybės atributas, nusakantis, kiek iš viso skirtingų sudedamųjų dalių (žr. 3.3.3.1 skirsnį) prisideda prie to, kad konkreti savybė būtų užtikrinama nagrinėjamame elemente.

Norint gauti atributo reikšmę, reikia suskaičiuoti visus netuščius langelius atsakomybių matricos savybės eilutėje (žr. 3.3.4.1 skirsnį). Tai yra, pritaikyti formulę:

$$f(y) = \sum_{i=1}^n 1, \text{ tiems } i, \text{ kuriems } Q(x_i, y) > 0, n > 0. \quad (2)$$

Čia y – savybė; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $Q(x, y)$ – sudedamosios dalies x atsakomybės už savybę y reikšmė.

Aukšta atributo reikšmė, artėjanti prie sudedamųjų dalių kiekio galėtų indikuoti tai, jog nag-

rinėjamame lygmenyje konkreti savybė yra nelokalizuojama ir ji turėtų būti perkelta į aukštesnį lygmenį. Nulinė reikšmė signalizuoja, kad atlikus dekompoziciją savybė nebus įgyvendinta.

3.3.5.2. Atsakingų dalių įverčių suma

Tai išvestinis kiekybinis savybės atributas, parodantis kaip stipriai elemento savybė realizuojama nagrinėjamame lygmenyje.

Norint gauti atributo reikšmę, reikia suskaičiuoti visų langelių reikšmių sumą atsakomybių matricos savybės eilutėje (žr. 3.3.4.1 skirsnį). Tai yra, taikyti formulę:

$$f(y) = \sum_{i=1}^n Q(x_i, y), n > 0. \quad (3)$$

Čia y – savybė; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $Q(x, y)$ – sudedamosios dalies x atsakomybės už savybę y reikšmė.

Šis atributas suteikia papildomos informacijos apie tai, koku intensyvumu savybė yra įgyvendinama nagrinėjamame lygmenyje. Atributo reikšmę galima traktuoti kaip galimą savybės įgyvendinimo bei palaikymo sudėtingumą. Pavyzdžiui, tarkime, kad savybę stipriai arba vidutiniškai realizuoja dauguma sudedamųjų dalių. Tokiu atveju, norint atlikti bet kokius pokyčius, susijusius su šia savybe arba už ją atsakingomis dalimis, reikės peržiūrėti arba įgyvendinti pokyčius visuose elemento sudedamosiose dalyse, turinčiose aukštą arba vidutinį atsakomybės įvertį.

3.3.5.3. Atsakingų dalių įverčių vidurkis

Tai išvestinis kiekybinis savybės atributas, parodantis, koks yra vidutinis už konkrečią savybę atsakingos sudedamosios dalies įvertis.

Norint gauti atributo reikšmę, reikia taikyti formulę, pasinaudojant (2) ir (3) formulėmis:

$$f(y) = \frac{\sum_{i=1}^n 1}{\sum_{i=1}^n Q(x_i, y)}, \text{ tiems } i, \text{ kuriems } Q(x_i, y) > 0, n > 0. \quad (4)$$

Čia y – savybė; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $Q(x, y)$ – sudedamosios dalies x atsakomybės už savybę y reikšmė.

Šis atributas gali suteikti informacijos apie dekomponavimo rezultato poveikį savybės sudėtingumui. Pavyzdžiui, aukšta reikšmė gali indikuoti apie tai, kad dauguma sudedamųjų dalių stipriai įgyvendina savybę, kas reikštų, kad savybės įgyvendinimas padidins sistemos sudėtingumą. Ir atvirkščiai, žema vidurkio vertė signalizuoja, jog didelė dalis sudedamųjų dalių savybę įgyvendina

paviršutiniškai.

Abiem atvejais architektui reikėtų peržiūrėti dekomponavimo rezultatą ir užtikrinti, kad savybės realizacija nėra išskaidyta į perteklinį kiekį elemento dalių ir bent viena dalis turi stiprią atsakomybę už savybės įgyvendinimą.

3.3.5.4. Alternatyvų savybių atributai

Savybių konkurencingumo verčių dalis pildoma tik tada, kai architektas tą patį nagrinėjamą elementą pasirenka dekomponuoti skirtingais būdais. Tokiu atveju šioje kortos dalyje atsидurs savybių konkurencingumo vertinimas, lyginantis gautas savybių atributų reikšmes su kitais būdais dekomponuoto elemento atributų reikšmėmis.

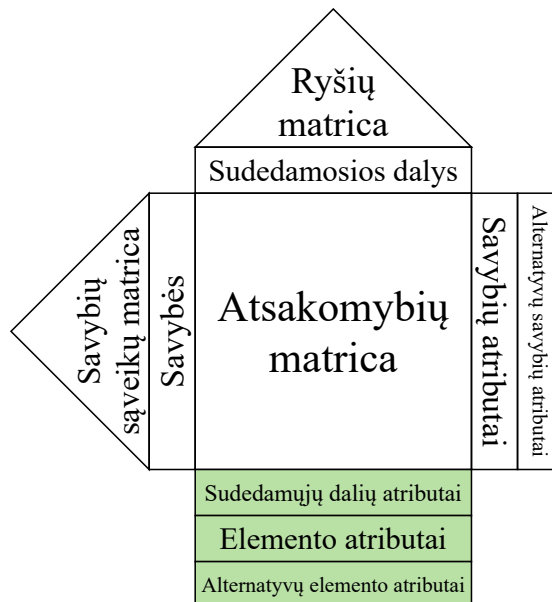
Pirma reikia užpildyti visus atributus iš skirsnių 3.3.5.1-3.3.5.3. Tada iš alternatyvių dekomponavimo būdų kortų importuojamos juose gautos atitinkamų savybių atributų reikšmės.

Lyginimas yra naudingas dėl to, kad galima atsekti alternatyvių dekomponavimo būdų poveikį kiekvienai individualiai nagrinėjamo elemento savybei. Šis būdas leidžia patogiai išdėstyti ir gretinti reikiamą informaciją sprendimų priėmimui. Tačiau DK metodas nepateikia būdo kaip vertinti lyginamą informaciją. Taip yra dėl to, kad siekiamo dekomponavimo rezultato poreikiai gali skirtis: galima siekti tiek aukšto, tiek žemo atsakomybių susiejimo laipsnio su savybėmis, todėl sprendimas paliekamas architekto kompetencijoms ir poreikiams.

3.3.6. Apatinė dekomponavimo kortos dalis

29 pav. paryškintos sudedamųjų dalių atributų grupės, elemento atributų grupės ir alternatyvų elemento atributų grupės kortos dalys, išdėstytos dekomponavimo kortos apačioje.

Visų pirma, apatinė dekomponavimo kortos dalis pradeda pildyti nuo sudedamųjų dalių atributų. Tai rinkinys kiekybinių atributų, skirtų suteikti daugiau informacijos apie sudedamąsias dalis, naudojant pasirinktą dekomponavimo būdą. Atributų reikšmės pildomos kiekvienai individualiai daliai atskirai. Tada yra užpildomi elemento atributai, skirti suteikti informacijos apie patį dekomponuojamą elementą. Elemento atributų reikšmės gaunamos atliekant visų sudedamųjų dalių atributų reikšmių sintezavimą.



29 pav. Apatinė dekomponavimo kortos dalis

3.3.6.1. Svarbos reitingas

Tai išvestinis kiekybinis sudedamosios dalies atributas, parodantis, kokia yra sudedamosios dalies svarba. Šis atributas susieja savybės prioriteto reitingo reikšmes (žr. 3.3.2.3 skirsnį) su sudedamosios dalies atsakomybių vertėmis (žr. 3.3.4.1 skirsnį) ir yra apskaičiuojamas taikant formulę:

$$f(x) = \sum_{j=1}^m (Q(x, y_j) \cdot P(y_j)), m > 0. \quad (5)$$

Čia x – sudedamoji elemento dalis; m – visų nagrinėjamo elemento savybių kiekis; $Q(x, y)$ – sudedamosios dalies x atsakomybės už savybę y reikšmė; $P(y)$ – savybės y prioriteto reitingo reikšmė.

Šis atributas gali padėti apsispręsti dėl sudedamosios dalies įgyvendinimo bei palaikymo prioriteto. Jeigu sudedamoji dalis atsakinga už svarbių savybių realizaciją, gali būti siekiama, kad tokios dalies kokybė būtų kuo aukštesnė. Taigi, esant aukštai svarbos reitingo atributo reikšmei, yra prasminga skirti proporcingo dydžio dekomponavimo pastangas sudedamajai daliai, siekiant sumažinti jos sudėtingumą.

3.3.6.2. Santykinė svarba

Tai išvestinis kiekybinis sudedamosios dalies atributas, nusakantis, kokia yra sudedamosios dalies svarba visų kitų dalių atžvilgiu, t. y. elemento dekompozicijos kontekste.

Norint apskaičiuoti atributo reikšmę, reikia taikyti formulę:

$$f(x) = \frac{g(x)}{\sum_{i=1}^n g(x_i)} \cdot 100\%, n > 0. \quad (6)$$

Čia x – sudedamoji dalis; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $g(x)$ – sudedamosios dalies x svarbos reitingas.

Esant baigtiniams ištekliams, galima susidurti su situacija, kai reikia apsiriboti kuriamos sistemos apimtimi. Tokiu atveju reikėtų rinktis, ką įtraukti į sistemos apimtį arba kurių dalių kūrimą pradėti pirmiausiai. Santykinė sudedamosios dalies svarba gali palengvinti šį pasirinkimą, nes atributą galima traktuoti kaip prioritetą. Kuo aukštesnė santykinė svarba, tuo daugiau reikalingų savybių sudedamoji dalis realizuos ir tuo daugiau pridėtinės vertės bus suteikta kuriamai sistemai.

Atributo reikšmę analogiškai galima traktuoti renkantis, kurias dalis toliau dekomponuoti tolimesniuose lygmenyse. Pavyzdžiui, nuspręsti, kad toliau bus dekomponuojamos tik tos elemento dalys, kurių svarbos reitingas yra aukštesnis nei 5%.

Tačiau neproporcingai aukštos atributo reikšmės galimai signalizuoja apie dideles ir sudėtingas dalis, kurių sukūrimui ir priežiūrai gali prireikti atitinkamai didelių resursų. Jeigu kuriant sistemą yra siekiama išvengti tokio tipo dalių, tuomet reikia peržiūrėti dekompoziciją ir pabandyti dekomponuoti elementą į dalis taip, kad jų santykinės svarbos atributų reikšmės būtų panašios.

3.3.6.3. Atsakomybių kiekis

Tai išvestinis kiekybinis sudedamosios dalies atributas, nusakantis kiekį savybių, už kurias sudedamoji dalis yra atsakinga.

Norint gauti atributo reikšmę, reikia suskaičiuoti visus netuščius langelius atsakomybių matricos sudedamosios dalies stulpelyje (žr. 3.3.4.1 skirsnį). Tai yra, pritaikyti formulę:

$$f(x) = \sum_{j=1}^m 1, \text{ tiems } j, \text{ kuriems } Q(x, y_j) > 0, m > 0. \quad (7)$$

Čia x – sudedamoji dalis; m – visų nagrinėjamo elemento savybių kiekis; $Q(x, y)$ – sudedamosios dalies x atsakomybės už savybę y reikšmė.

Atsakomybių kiekis yra svarbus dekompozicijos vertinimui, nes leidžia atpažinti sudėtingas elemento dalis. Jeigu atributo reikšmė artėja prie savybių kiekio m , tai parodo, kad sudedamoji dalis turės aukštą sudėtingumą, nes bus atsakinga už daugumą savybių. Kai atributo reikšmė yra 0, tai reiškia, kad elemento dalis yra perteklinė ir nekuria pridėtinės vertės. Abiem atvejais reikia

peržiūrėti dekomponavimo rezultatą ir įvertinti, ar nėra tinkamesnio būdo išskaidyti elementą į dalis taip, kad būtų išvengta minėtų tendencijų.

3.3.6.4. Atsakomybių verčių suma

Tai išvestinis kiekybinis sudedamosios dalies atributas, parodantis, kokio dydžio atsakomybę dekomponuotas elementas turi nagrinėjamame lygmenyje.

Norint gauti atributo reikšmę, reikia suskaičiuoti visų langelių reikšmių sumą atsakomybių matricos sudedamosios dalies stulpelyje (žr. 3.3.4.1 skirsnį). Tai yra, taikyti formulę:

$$f(x) = \sum_{j=1}^m Q(x, y_j), m > 0. \quad (8)$$

Čia x – sudedamoji dalis; m – visų nagrinėjamo elemento savybių kiekis; $Q(x, y)$ – sudedamosios dalies x atsakomybės už savybę y reikšmė.

Šis atributas papildo atsakomybių kiekio atributą (žr. 3.3.6.3 skirsnį), identifikuodamas tendencijas, kurių kuriant sistemą yra norima išvengti. Sakykime, kad nagrinėjamas elementas turi daug savybių, o sudedamoji dalis yra atsakinga vos už kelias savybes (1 arba 2). Tokiu atveju, ieškodami tendencijų naudojant atsakomybių kiekio atributą (žr. 3.3.6.3 skirsnį), nerastume nieko neįprasto.

Tačiau, gali būti taip, kad sudedamosios dalies, turinčios 1 arba 2 atsakomybes, verčių suma atitinkamai taip pat bus 1 arba 2. Tai indikuoja apie silpnas atsakomybes, kurios galimai nėra būtinos tam, kad atitinkamos savybės būtų pilnai realizuojamos elemente. Architektui reikėtų atsakyti į klausimą, ar įmanoma dekomponuoti elementą be šių dalių. Elemento dalis galutiniame dekomponavimo rezultate turi būti arba palikta, nes yra būtina, arba pašalinta, nes tai sumažintų elemento sudėtingumą.

3.3.6.5. Atsakomybių verčių vidurkis

Tai išvestinis kiekybinis sudedamosios dalies atributas, parodantis, koks yra vidutinis sudedamosios dalies atsakomybės įvertis.

Norint gauti atributo reikšmę, reikia taikyti formulę, pasinaudojant (7) ir (8) formulėmis:

$$f(x) = \frac{\sum_{j=1}^m Q(x, y_j)}{\sum_{j=1}^m 1}, \text{ tiems } j, \text{ kuriems } Q(x, y_j) > 0, m > 0. \quad (9)$$

Čia x – savybė; m – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $Q(x, y)$ – sudedamosios

dalies x atsakomybės už savybę y reikšmė.

Atributas yra naudingas dėl to, kad leidžia įvertinti sudedamųjų dalių pobūdį. Pavyzdžiui, atributo reikšmei esant mažesnei negu 2, galima užtikrintai tvirtinti, kad didžioji dalis sudedamosios dalies atsakomybių yra silpnos⁷. Tai gali indikuoti apie pagalbines dalis, kurios prisideda prie didelės dalies skirtingų atsakomybių, tačiau savybių užtikrinimas kritiškai nenukentėtų, jeigu dalys ir nebūtų realizuotos. Siekiant užtikrinti kuo mažesnę elemento sudėtingumą, reikėtų riboti tokių dalių egzistavimą.

Tačiau jeigu atributo reikšmė yra didesnė negu 6, galima užtikrintai tvirtinti, kad didžioji sudedamosios dalies atsakomybių yra stiprios⁸. Tai sudėtingos dalys, kuriose galimai sukonzentruotos nagrinėjamo elemento atsakomybės. Tokių dalių egzistavimą taip pat yra prasminga riboti, nes tai padidina sudedamųjų dalių, o kartu ir elemento sudėtingumą.

3.3.6.6. Ryšių kiekis

Tai išvestinis kiekybinis sudedamosios dalies atributas, parodantis, kiek iš viso yra kitų elemento dalių, su kuriomis konkreti sudedamoji dalis yra tiesiogiai susijusi.

Norint gauti atributo reikšmę, reikia suskaičiuoti visus netuščius langelius ryšių matricos sudedamosios dalies srityje (žr. 3.3.3.3). Tai yra, taikyti formulę:

$$f(x) = \sum_{i=1}^n 1, \text{ tiems } i, \text{ kuriems } x \neq x_i \text{ ir } C(x, x_i) > 0, n > 0. \quad (10)$$

Čia x, x_i – sudedamosios dalys; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $C(x, x_i)$ – sudedamųjų dalių x ir x_i susikirtimo reikšmės ryšių matricoje (žr. 3.3.3.3).

Kiekvienas papildomas ryšys sukuria priklausomybę tarp dviejų elemento dalių. Be to, svarbu užtikrinti, kad kiekviena priklausomybė būtų stabili, t. y. ryšys nenutrūktų, o transakcijos tarp jų vyktų taip, kaip numatyta. Taigi, atliekant bet kokias elemento dalių modifikacijas arba išplėtimus, reikia patikrinti, ar ryšiai po pakeitimų išliko korektiški. Kuo daugiau sudedamoji dalis turės ryšių, tuo daugiau pastangų reikės skirti tų ryšių priežiūrai ir tuo inertiškesnė bus elemento dalis.

3.3.6.7. Ryšių verčių suma

Tai išvestinis kiekybinis sudedamosios dalies atributas, parodantis, kokio stiprumo ryšį dalis turi su visomis kitomis elemento dalimis.

⁷Didesnė atributo reikšmė nebūtinai reiškia, kad silpna atsakomybė nėra dominuojanti.

⁸Tam tikrais atvejais stipri atsakomybė gali dominuoti esant ir mažesnei atributo reikšmei.

Norint gauti atributo reikšmę, reikia suskaičiuoti visų langelių sumą ryšių matricos sudedamosios dalies srityje (žr. 3.3.3.3). Tai yra, taikyti formulę:

$$f(x) = \sum_{i=1}^n C(x, x_i), \text{ tiems } i, \text{ kuriems } x \neq x_i, n > 0. \quad (11)$$

Čia x, x_i – sudedamosios dalys; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $C(x, x_i)$ – sudedamųjų dalių x ir x_i susikirtimo reikšmės ryšių matricoje (žr. 3.3.3.3).

Šis atributas nėra naudojamas elemento dalių tendencijų paieškoms, tačiau jis yra reikalingas ryšių verčių vidurkio reikšmės gavimui (žr. 3.3.6.8 skirsnį).

3.3.6.8. Ryšių verčių vidurkis

Tai išvestinis kiekybinis sudedamosios dalies atributas, parodantis, kokio stiprumo ryšį dalis vidutiniškai turi su kitomis elemento dalimis.

Norint gauti atributo reikšmę, reikia taikyti formulę, pasinaudojant (10) ir (11) formulėmis:

$$f(x) = \frac{\sum_{i=1}^n C(x, x_i)}{\sum_{i=1}^n 1}, \text{ tiems } i, \text{ kuriems } x \neq x_i \text{ ir } C(x, x_i) > 0, n > 0. \quad (12)$$

Čia x, x_i – sudedamosios dalys; n – visų nagrinėjamo elemento sudedamųjų dalių kiekis; $C(x, x_i)$ – sudedamųjų dalių x ir x_i susikirtimo reikšmės ryšių matricoje (žr. 3.3.3.3).

Šis atributas gali suteikti daugiau informacijos apie elemento dalis. Pavyzdžiui, ar sudedamojoje dalyje dominuoja stiprūs ar silpni ryšiai. Jeigu atributo reikšmė yra didesnė negu 2, tai reiškia, kad vyrauja stiprūs ryšiai. Jeigu atributo reikšmė mažesnė negu 2, tuomet vyrauja silpni ryšiai.

Tam, kad sudedamosios dalies sudėtingumas būtų valdomas, reikėtų vengti dominuojančių stiprių ryšių. Pagrindinis būdas to išvengti – atpažinti stiprių ryšių tendenciją ir peržiūrėti dekomponavimo rezultata. Gali būti, kad elementą galima dekomponuoti išvengiant dominuojančio stipraus ryšio.

3.3.6.9. Sudedamųjų dalių kiekis

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis, į kiek dalių esamajame lygmenyje yra dekomponuotas elementas. sudedamųjų dalių kiekis žymimas raide n . Bendruoju atveju didesnis sudedamųjų dalių kiekis reiškia didesnę elemento sudėtingumą, nes padaugėja galimų ryšių kombinacijų ir reikiamų pastangų papildomoms dalims sukurti ir palaikyti.

Tiesa, realybėje mažesnis sudedamųjų dalių kiekis nulemia tankesnę atsakomybių pasiskirstymą dalyse ir daugiau dekomponavimo lygmenų dėl vidutiniškai stambesnio granuliarumo kiekviename lygmenyje. Tačiau esant vienodomis sąlygoms, sistema su didesniu dalių kiekiu atitinkamai bus sudėtingesnė.

3.3.6.10. Svarbos reitingų suma

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, nusakantis nagrinėjamo elemento apibendrinantį svarbos reitingą.

Norint gauti atributo reikšmę, reikia suskaičiuoti visų elemento dalių svarbos reitingų reikšmių sumą (žr. 3.3.6.1). Tai yra, taikyti formulę:

$$\alpha = \sum_{i=1}^n f(x_i), n > 0. \quad (13)$$

Čia α – nagrinėjamo elemento svarbos reitingų sumos atributas; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x svarbos reitingas (žr. (5) formulę).

Kuo aukštesnė svarbos reitingų sumos atributo reikšmė, tuo daugiau dėmesio ir pastangų reikės skirti pačio elemento kūrimui bei priežiūrai. Tuo pačiu lygmenyje, kuriame yra nagrinėjamas elementas, kartu egzistuoja ir kiti elementai, turintys savo svarbos sumos atributus, ir kurie daro įtaką aukštesnio lygmens elementų svarbai. Todėl, jeigu architektas bus neapdairus ir pasirinks dekomponavimo būdus su labai aukštomis svarbos sumos atributo reikšmėmis, galutinis sistemos sudėtingumas bus aukštesnis, negu pasirinkus žemesnio svarbos reitingo alternatyvas.

Atributo reikšmė tiesiogiai priklauso nuo savybių santykinės svarbos, sudedamųjų dalių atsakomybių kiekio bei jų stiprumo. Architektas, dekomponuodamas elementą į dalis, gali kontroliuoti atsakomybių kiekį ir stiprumą, nes savybių santykinė svarba yra fiksuotas dydis. Tačiau nebūtinai reikia siekti žemiausios įmanomos svarbos reitingų sumos reikšmės. Tokio siekio pasekmė gali tapti paviršutiniškai įgyvendinamos savybės. Todėl svarbu, kad būtų užtikrinama pusiausvyra tarp savybių užtikrinimo bei bendros elemento svarbos reitingo.

3.3.6.11. Svarbos reitingų mediana

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis visų sudedamųjų dalių svarbos reitingų medianą.

Norint gauti atributo reikšmę, reikia suskaičiuoti visų elemento dalių svarbos reitingų media-

ną. Tai yra, taikyti formulę:

$$\alpha = \text{med}(f(x_1), \dots, f(x_n)), n > 0. \quad (14)$$

Čia α – nagrinėjamo elemento svarbos reitingų mediana; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x svarbos reitingas (žr. (5) formulę).

Kuriant DK metodą buvo svarstoma rinktis tarp vidurkio ir medianos atributo. Galiausiai nuspręsta likti prie medianos reikšmės. Autoriaus manymu, toks sprendimas leidžia patikimiau nustatyti vidurinę sudedamųjų dalių svarbos reitingo reikšmę dėl galimų vidurkio iškreipimų. Šie iškreipimai susidaro esant vienai ar kelioms neproporcingai aukštą svarbą kitų dalių atžvilgiu turinčioms elemento dalims (plačiau apie iškreipimus žiūrėti 3.3.6.12 skirsnyje).

Atributo reikšmė gali būti naudinga lyginant kelias dekomponavimo alternatyvas tarpusavyje. Esant aukštesnei atributo reikšmei, yra tikėtina, kad dekomponuotame elemente esančios dalies įgyvendinimui ir priežiūrai reikės vidutiniškai daugiau pastangų bei dėmesio, negu pritaikius būdą su mažesne atributo reikšme. Architektas turėtų dėti pastangas, kad atlikus dekompoziciją, svarbos reitingų mediana būtų kuo žemesnė. Tačiau, kaip ir 3.3.6.10 skirsnio atveju, reikia rasti balansą tarp savybių užtikrinimo ir sudedamųjų dalių svarbos reitingo.

3.3.6.12. Santykinės svarbos rangas

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis skirtumą tarp didžiausios ir mažiausios visų sudedamųjų dalių santykinės svarbos reikšmės.

Norint gauti atributo reikšmę, reikia iš didžiausios visų elemento dalių santykinės svarbos reitingo reikšmės atimti mažiausią reikšmę. Tai yra, taikyti formulę:

$$\alpha = \max_{\forall i \in n} f(x_i) - \min_{\forall i \in n} f(x_i), [n] = \{1, \dots, n\}. \quad (15)$$

Čia α – nagrinėjamo elemento santykinės svarbos rangas; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x santykinės svarbos reikšmė (žr. (6) formulę).

Santykinės svarbos rango atributas gali parodyti, koks yra elemento dalių atsakomybių pasiskirstymas. Aukšta atributo reikšmė indikuoja, kad du ekstremumai, t. y. svarbiausia ir mažiausiai svarbi sudedamoji dalis, turi neproporcingas atsakomybes. Atributo reikšmė turi būti kuo mažesnė, jeigu siekiama suvienodinti sudedamųjų dalių svarbą nagrinėjamame elemente ir atvirkščiai, jeigu norima sukcentruoti daugumą elemento savybių į vieną ar kelias dalis.

3.3.6.13. Atsakomybių kiekis

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis visų sudedamųjų dalių atsakomybių tankumą elemente.

Norint gauti atributo reikšmę, reikia suskaičiuoti, kiek iš viso yra užpildytų atsakomybių matricos laukų (žr. 3.3.4.1 skirsnį). Tai yra, taikyti formulę:

$$\alpha = \sum_{i=1}^n f(x_i), n > 0. \quad (16)$$

Čia α – nagrinėjamo elemento atsakomybių kiekis; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x atsakomybių kiekis (žr. (7) formulę).

Tai yra atributas, leidžiantis prognozuoti elemento sudėtingumą. Atributo reikšmė priklauso nuo elemento savybių bei sudedamųjų dalių kiekio, taip pat nuo atsakomybių matricos įverčių. Taigi, architektas gali kontroliuoti tik elemento dalis: į kiek ir kokias dalis bus dekomponuojama ir kokios atsakomybės joms bus priskirtos.

Kiekviena papildoma atsakomybė apsunkina elemento kūrimą ir priežiūrą. Sukūrus sistemą, reikia garantuoti, kad kiekviena atsakomybė būtų užtikrinama. Be to, esant savybių pokyčiams, reikalinga atlikti pakeitimus sudedamosiose dalyse, turinčiose atsakomybę.

Labai tankus elemento atsakomybių pasiskirstymas (kai α artėja prie $n \cdot m$; m – tai elemento savybių kiekis) indikuoja apie dekomponavimo būdą, kuriame gausu milžiniškų dalių, nėra išsiskiriančių dalių su aiškėmis, išreikštinai apibrėžtomis atsakomybių ribomis. Esant šiai tendencijai, architektas turėtų peržiūrėti dekompoziciją ir ieškoti tinkamesnių alternatyvų.

3.3.6.14. Atsakomybių verčių suma

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis visų sudedamųjų dalių atsakomybių intensyvumą elemente.

Norint gauti atributo reikšmę, reikia suskaičiuoti visų atsakomybių matricos laukų įverčių sumą (žr. 3.3.4.1 skirsnį). Tai yra, taikyti formulę:

$$\alpha = \sum_{i=1}^n f(x_i), n > 0. \quad (17)$$

Čia α – nagrinėjamo elemento atsakomybių verčių suma; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x atsakomybių verčių suma (žr. (8) formulę).

Šis atributas papildo atsakomybių kiekio atributą. Gali egzistuoti tokie scenarijai, kai dviejų dekomponavimo būdų atsakomybių kiekio atributai sutampa arba yra labai panašūs. Tokiu atveju atsakomybių verčių sumos reikšmė suteikia daugiau informacijos apie tai, kuria dekomponavimo alternatyva pasinaudojus nagrinėjamo elemento sudėtingumas galimai sumažėtų.

Pastangų kiekis, esant skirtingiems atsakomybių tipams, gali skirtis kartais. Todėl atsakomybių verčių sumos atributas, siekiant prognozuoti elementui skiriamų pastangų kiekį, gali būti naudingesnis, negu atsakomybių kiekio atributas. Jeigu atributo reikšmė yra mažesnė negu $9n$, tokiu atveju yra rizika, kad bent viena iš elemento savybių bus įgyvendinta paviršutiniškai. Tai yra, egzistuos savybė arba savybės, už kurias atsakingos sudedamosios dalys neturės stiprios atsakomybės.

3.3.6.15. Atsakomybių verčių vidurkio rangas

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis skirtumą tarp didžiausios ir mažiausios visų sudedamųjų dalių atsakomybių verčių vidurkio reikšmės.

Norint gauti atributo reikšmę, iš didžiausios visų elemento dalių atsakomybių verčių vidurkio reikšmės reikia atimti mažiausią reikšmę. Tai yra, taikyti formulę:

$$\alpha = \max_{\forall i \in n} f(x_i) - \min_{\forall i \in n} f(x_i), [n] = \{1, \dots, n\}. \quad (18)$$

Čia α – nagrinėjamo elemento atsakomybių verčių vidurkio rangas; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x atsakomybių verčių vidurkis (žr. (9) formulę).

Tai atributas, leidžiantis atpažinti tokias nagrinėjamo elemento dalis, kurios turi neproporcingai stiprias arba silpnas atsakomybes. Pavyzdžiui, elemento dalis X gali turėti 10 silpnų atsakomybių. Tuo tarpu dalis Y gali turėti tik vieną, bet stiprią atsakomybę. Tada Y dalies rango atributo reikšmė bus labai aukšta.

Kuriant sistemą gali iškilti poreikis vengti kraštutinumų ir siekti harmoningo atsakomybių vidutinio stiprumo paskirstymo sudedamosioms dalims. Kai α reikšmė artėja prie 8, tai reiškia, kad elemente yra sudedamųjų dalių su vien tik stipriomis, ir su vien tik silpnomis atsakomybėmis. Norint pasiekti atsakomybių harmoniją nagrinėjamame elemente, reikia užtikrinti, kad atsakomybių verčių vidurkio rangas būtų kuo artimesnis reikšmei 0.

3.3.6.16. Ryšių kiekis

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis bendrą visų sudedamųjų dalių tarpusavio ryšių tankumą elemente.

Norint gauti atributo reikšmę, reikia suskaičiuoti, kiek iš viso yra užpildytų ryšių matricos laukų (žr. 3.3.3.3 skirsnį). Tai yra, taikyti formulę:

$$\alpha = \frac{\sum_{i=1}^n f(x_i)}{2}, n > 0. \quad (19)$$

Čia α – nagrinėjamo elemento ryšių kiekis; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x ryšių su kitomis dalimis kiekis (žr. (10) formulę). Dalyba iš 2 atliekama todėl, kad kiekvienos sudedamosios dalies ryšys yra abipusis. Tai yra, skaičiuojant visų dalių kiekį sumuojant formulę $f(x)$, yra įtraukiami ryšiai tiek tarp x_1 ir x_2 , tiek tarp x_2 ir x_1 dalių, nors tai yra vienodi ryšiai.

Ryšių kiekio atributas leidžia atpažinti 3.3.3.3 skirsnyje aprašytų A ir B tendencijų variantus. α reikšmei artėjant prie $\frac{n(n-1)}{2}$, reiškia, kad didžioji dalis sudedamųjų dalių bus priklausomos nuo daugumos kitų dalių. Ši tendencija yra negatyvi ir signalizuoja, kad dauguma sudedamųjų dalių bus inertiškos ir sunkiai plečiamos.

Lyginant dekomponavimo alternatyvas, vienas iš pasirinkimų nulemiančių argumentų gali būti aukšta ryšių kiekio atributo reikšmė. Architektas gali įvertinti, kad aukšta atributo reikšmė gali sudaryti per daug rizikų nagrinėjamo elemento sukūrimui bei tolimesnei priežiūrai. Užtikrinus žemesnę atributo reikšmę būtų sumažinamas elemento sudėtingumas ir galimos rizikos.

3.3.6.17. Ryšių verčių suma

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis visų sudedamųjų dalių tarpusavio ryšių intensyvumą elemente.

Norint gauti atributo reikšmę, reikia suskaičiuoti, visų ryšių matricos laukų įverčių sumą (žr. 3.3.3.3 skirsnį). Tai yra, taikyti formulę:

$$\alpha = \frac{\sum_{i=1}^n f(x_i)}{2}, n > 0. \quad (20)$$

Čia α – nagrinėjamo elemento ryšių verčių suma; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x ryšių su kitomis dalimis verčių suma (žr. (11) formulę). Dalyba iš 2 atliekama dėl priešasčių, pristatytų (19) formulės paaiškinime.

Tai atributas, papildantis ryšių kiekio atributą (žr. 3.3.6.16 skirsnį). Gali būti, taip, kad lyginamų dekomponavimo būdų ryšių kiekio atributai sutampa arba reikšmės yra labai artimos. Esant tokiam scenarijui, architektui reikėtų atkreipti dėmesį į elemento ryšių verčių sumos atributą. Aukštesnė atributo vertė indikuoja, kad ryšiai tarp sudedamųjų dalių yra intensyvesni ir sudėtingesni, todėl elemento kūrimui gali reikėti daugiau pastangų bei kilti daugiau rizikų.

3.3.6.18. Ryšių kiekio rangas

Tai išvestinis kiekybinis nagrinėjamo elemento atributas, parodantis skirtumą tarp didžiausios ir mažiausios visų sudedamųjų dalių ryšių kiekio reikšmės.

Norint gauti atributo reikšmę, iš didžiausios visų elemento dalių ryšių kiekio reikšmės reikia atimti mažiausią reikšmę. Tai yra, taikyti formulę:

$$\alpha = \max_{\forall i \in n} f(x_i) - \min_{\forall i \in n} f(x_i), [n] = \{1, \dots, n\}. \quad (21)$$

Čia α – nagrinėjamo elemento ryšių kiekio rangas; n – nagrinėjamo elemento sudedamųjų dalių kiekis; $f(x)$ – sudedamosios dalies x ryšių kiekio reikšmė (žr. (10) formulę).

Atributo reikšmė gali suteikti įžvalgų apie neproporcingai milžiniškas arba rizikingas elemento dalis, pavaizduotas 3.3.3.3 skirsnio C tendencijoje. Pavyzdžiui, elementas gali būti dekomponuojamas taip, kad savyje talpintų dalį, žinančią apie visas kitas elemento dalis ir koordinuojančią transakcijas tarp jų. Toks dalies egzistavimas sukelia rimtų rizikų, nes turi aukštą klaidos kainą. sudedamajai daliai tapus nestabilios būsenos, būtų paveiktos visos sudedamosios dalys, taigi, viso elemento būseną taip pat taptų nestabili.

Taigi, jeigu α reikšmė priartėja prie $(n - 1)$, architektui reikėtų atkreipti dėmesį į galimą šalutinį sudedamųjų dalių poveikį. Jeigu atributo reikšmė artėja prie 0, tai reiškia, jog dekomponavus elementą pavyko pasiekti harmoniją ir išvengti rizikingų dalių.

3.3.6.19. Alternatyvų elemento atributai

Sudedamųjų dalių konkurencingumo verčių dalis pildoma tik tada, kai architektas tą patį nagrinėjamą elementą pasirenka dekomponuoti skirtingais būdais. Tokiu atveju šioje kortos dalyje atsidurs elemento konkurencingumo vertinimas, lyginantis sintezuotus visų elemento dalių atributus su kitais būdais dekomponuoto elemento atributų reikšmėmis.

3.3.5.4 skirsnyje aprašoma, kaip nagrinėjamo elemento dekomponavimo būdai yra lyginami savybių atžvilgiu. Tai yra, kokį poveikį pasirinktas dekomponavimo būdas turi individualioms

savybėms. Tačiau sudedamųjų dalių atžvilgiu dekomponavimo būdų lyginimo principas skiriasi. Nepriklausomai nuo pasirinkto dekomponavimo būdo, savybių sąrašas nesiskiria. Tuo tarpu dekomponavimo dalių sąrašas, priklausomai nuo dekomponavimo būdo, neišvengiamai skirsis.

Gretinant nagrinėjamo elemento sintezuotus atributus, galima lengviau rasti trūkumus bei privalumus tarp dekomponavimo alternatyvų. Tačiau DK metodas nepateikia vienareikšmiško atsakymo, kuris dekomponavimo būdas geresnis dėl atributų įvairovės. Esminį sprendimą, kurią alternatyvą pasirinkti, turi priimti pats architektas, ekspertiškai įvertinęs kuriamos sistemos kontekstą ir įrankio pagalba gautų atributų reikšmes.

3.4. Dekomponavimo kortų metodo apibendrinančios pastabos

Tolimesniuose poskyriuose bus pateiktos baigiamosios DK metodo pastabos: metodo ribojimai ir galimos tolimesnės plėtros kryptys.

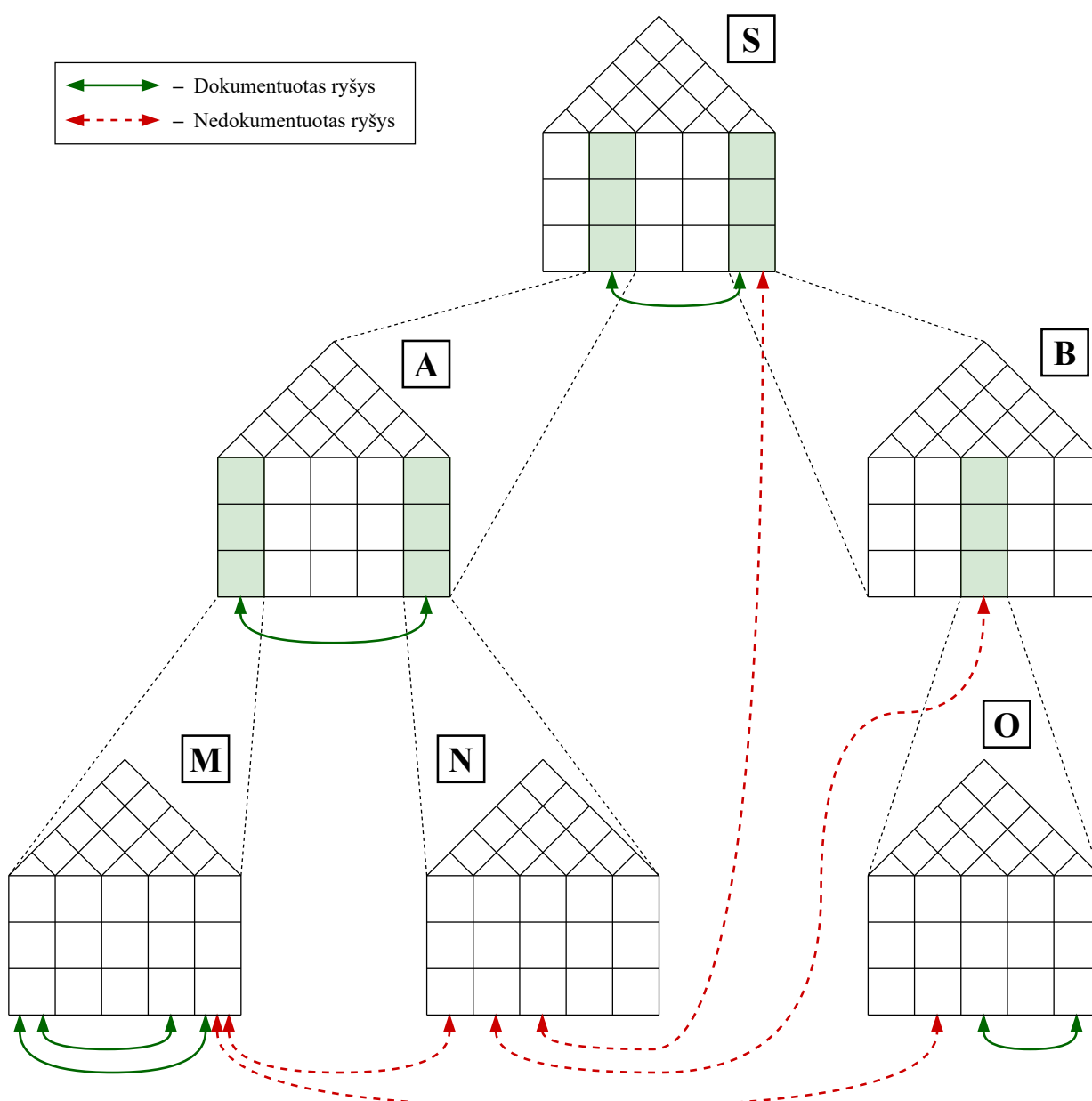
3.4.1. Dekomponavimo kortų metodo ribojimai

DK metodas ir jį realizuojantis įrankis turi tam tikrų ribojimų, išvardintų žemiau:

1. DK metodas leidžia stebėti, kaip atliekami dekomponavimo sprendimai daro įtaką nagrinėjamo elemento kiekybinių atributų pokyčiui. Tačiau atributai yra lokalizuojami tik nagrinėjamame elemente. Todėl nėra galimybės pamatyti, kaip kiekviena alternatyva ar sprendimas daro įtaką globaliu mastu. Tai yra, stebėti sprendimų poveikį aukštesniuose lygmenyse esantiems elementams ir jų sudėtingumui iki pat pirmojo sistemos lygmens.
2. DK metodas nenumato galimybės įvertinti, koku ryšiu nagrinėjamo elemento sudedamosios dalys yra susietos su kitų elementų dalimis tame pačiame lygmenyje. Taip pat nėra numatyta, kaip įvertinti dalių ryšio su aukštesnio arba žemesnio lygmens atominėmis dalimis. Tai reiškia, kad visi sudedamųjų dalių ryšiai bus fiksuojami lokaliai, nagrinėjamo elemento kontekste. Tokiu būdu yra prarandama dalis svarbios informacijos, kuri leistų tiksliau įvertinti sudedamųjų dalių sudėtingumą. Pavyzdys pateiktas 30 pav.
3. Dėl spartaus kortų augimo elementų dekomponavimas giliuose lygmenyse gali sueikvoti per daug resursų – kaštai gali būti didesni negu gaunama nauda. Sužinoti, kiek kortų reikės pildyti, galima atliekant skaičiavimus:

$$\beta = a \cdot \sum_{k=1}^n b^{k-1}. \quad (22)$$

Čia β – reikalingų užpildyti kortų kiekis; a – kiek vidutiniškai kiekvienam elementui yra užpildoma alternatyvių dekomponavimo kortų; b – vidutinis kiekviename elemente esančių sudedamųjų dalių kiekis; n – dekomponavimo lygmenų kiekis. Šis skaičiavimas yra apytikris, kadangi tiek alternatyvų, tiek elemente esančių sudedamųjų dalių kiekis gali labai skirtis. Tačiau pateikta skaičiavimo formulė leidžia suprasti, nuo kokių dydžių priklauso papildomų kortų augimo sparta bei prognozuoti dekomponavimui skiriamų pastangų kiekį. Kortų kiekio dydžiui didelę įtaką turi lygmenų kiekis, nes nuo jo priklauso augimo laipsnio rodiklis. Tačiau lygmenų kiekis yra ribotas, todėl spartus augimas yra kontroliuojamas. T. y. kortų augimas sustoja pasiekus atominės sudedamąsias sistemos elementų dalis.



30 pav. Sudedamųjų dalių ryšių dokumentavimas skirtingų lygmenų kortose

3.4.2. Dekomponavimo kortų metodo plėtojimo kryptys

Didelė dalis plėtojimo kryptių kyla dėl esamų ribojimų. Toliau pateikiamos sritys, kuriose būtų naudinga tęsti tyrimus, kurie leistų pagerinti DK metodą ir jo praktinį taikymą:

1. Sudedamųjų dalių globalaus poveikio visoms dalims matavimas. Būtų prasminga ieškoti būdų, kaip dekomponavus elementą, gebėti sužinoti jo dalių ryšius su visais sistemos elementais. Tai leistų gauti papildomos informacijos apie dalių sudėtingumą. Į DK metodą įtraukus globalių ryšių paiešką, dekomponavimo proceso pastangos ir sudėtingumas smarkiai išaugtų. Todėl reikia ieškoti būdų, kurie leistų paprasčiau fiksuoti dalių tarpusavio ryšius.
2. DK metodas neapima dekomponavimo dalių, esančių žemesniuose lygmenyse, poveikio visai sistemai. Galėtų būti įvestas kaupiamasis atributas arba atributų rinkinys, kuris iš žemiausių dekomponavimo lygmenų perneštų skaitines atributų vertes į aukštesnius lygmenis. Duomenų sintezavimas būtų vykdomas iki pat aukščiausio, sistemos lygmens, kuriame matytųsi visos sistemos sudėtingumo įvertis. Toks kaupiamasis atributas turėtų leisti stebėti dekomponavimo sprendimų įtaką atributams, o tai architektui leistų greičiau pasirinkti tinkamesnį dekomponavimo būdą.
3. DK metodą realizuojančio įrankio automatizavimas specializuotoje programinėje įrangoje. Kadangi įrankis parengtas „Microsoft Excel“ programinės įrangos aplinkoje, norint automatizuoti tam tikrus metodo aspektus, atsiranda techninių suvaržymų. Kuriant originalią programinę įrangą, skirtą DK metodui įgyvendinti, būtų galima automatizuoti tokius aspektus, kaip: kortos dalių pildymo eiliškumo užtikrinimą, tendencijų atpažinimą, laukų validacijas, dekomponavimo alternatyvų importavimą, neriboto kiekio sudedamųjų dalių ir savybių pridėjimą ir t. t. Be to, identifikavus pakankamai aukštą metodo kuriamą pridėtinę vertę, įrankio sukūrimas užtikrintų paprastesnę produkto komercializaciją.
4. Dekomponavimo kortų spartaus augimo optimizavimas. Norint išvengti spartaus kortų augimo, efektyviausia yra dekomponuoti sistemą į kelis lygmenis. Esant poreikiui dekomponuoti elementus į dalis žemesniuose lygmenyse, būtų galima imtis optimizavimo priemonių. Pavyzdžiui, dekomponuoti tik pačius svarbiausius elementus. Tačiau pasiūlytas būdas neišsprendžia problemos, nes šitokiu būdu būtų prarandama nemaža dalis informacijos. Neatmetama galimybė, kad sistemų architektui gali reikėti dekomponuoti visus elementus iki giliausio programinio lygmens su atominėmis dalimis. Tokiu atveju architektas negalėtų efektyviai naudoti DK metodo.
5. Sudedamųjų dalių integravimas į elementus. Šiame darbe buvo apsiribota dekomponavimo

procesu. Tačiau neatmetama galimybė, kad dekomponavus sistemą iki jos atominių dalių, būtų galima atlikti priešingą dekomponavimui veiksmą – integravimą. Integravimo metu būtų galima skaičiuoti kaupiamuosius atributus, paminėtus 2-ame plėtojimo kryptį punkte. Taip pat modifikuoti atliktus dekomponavimo sprendimus atliekant sudedamųjų dalių, turinčių aukštus tarpusavio ryšius, grupavimą (angl. *clusterization*) į elementus.

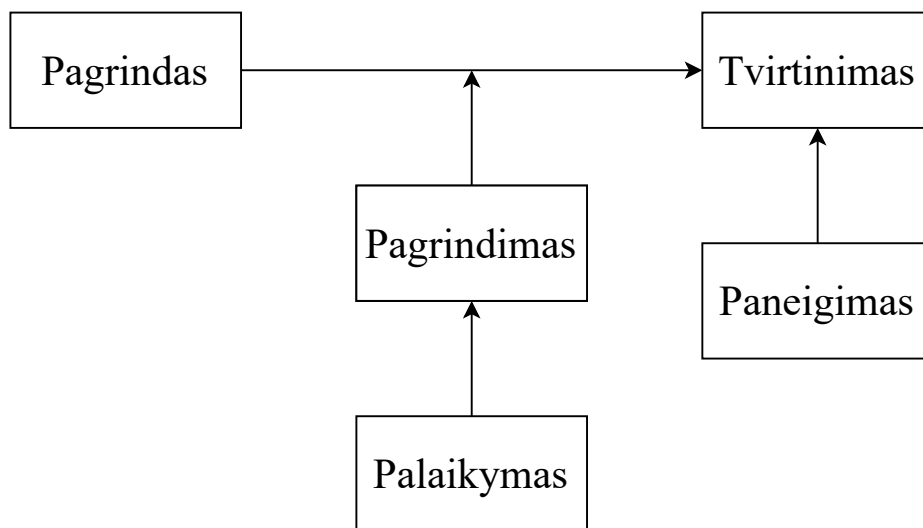
4. Dekomponavimo kortų metodo taikymas

Šiame skyriuje pateikiamas DK metodo argumentų pagrįstumas ir taikymo pavyzdys naudojant sukonstruotą metodą realizuojančią praktinę priemonę.

4.1. Argumentų pagrįstumas

Tiriamąjį darbo eigą buvo sukonstruota teorinė dalis, t. y. DK metodas ir jo taikymo nurodymai (žr. 3 skyrių). Tačiau apsiriboti teoriniu metodu neužtenka. Norint įsitikinti, kad sukurta teorija yra tinkama, ją reikia pagrįsti praktiniais eksperimentais. Vienas iš galimų pagrindimo būdų yra teorinio modelio pritaikymas praktikoje. Pavyzdžiui, DK metodo taikymas vienos arba kelių įmonių sistemų kūrimo procese. Šitaip būtų galima analizuoti gautus rezultatus, atlikti palyginimus ir apklausti architektus, kurie, remdamiesi savo ekspertinėmis žiniomis, galėtų įvertinti DK metodą.

Tačiau šis eksperimentinis būdas tiriamojo darbo apimtyje yra sunkiai įgyvendinamas dėl resursų trūkumo. Visų pirma, reikėtų surasti įmones ir jų specialistus, kurie sistemos kūrimo metu sutiktų eksperimentuoti naudojant naujai sukurtą ir dar neįrodyto efektyvumo DK metodą. Be to, būtų būtina stebėti sistemą ir rinkti duomenis jos viso kūrimo gyvavimo ciklo metu. Galiausiai, užbaigus sistemos stebėjimą, būtų privalu išanalizuoti sukauptus duomenis. Visa tai galėtų užtrukti iki kelių metų, o tai reikalautų per daug resursų magistro baigiamajam darbui atlikti.

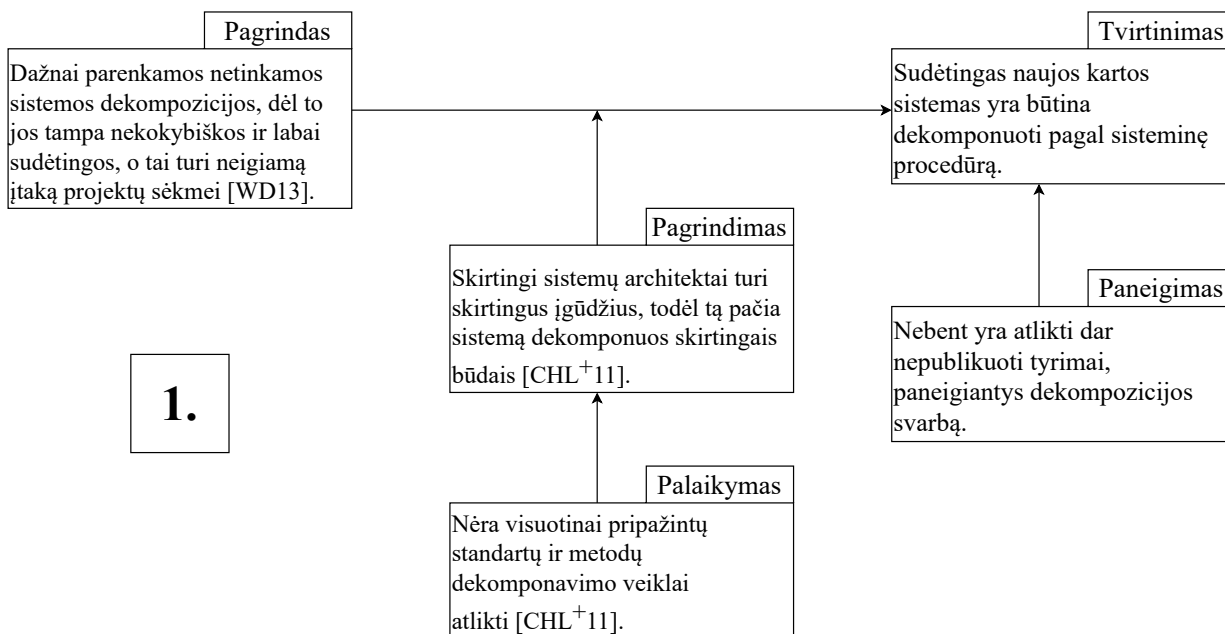


31 pav. Toulmino argumentavimo schema

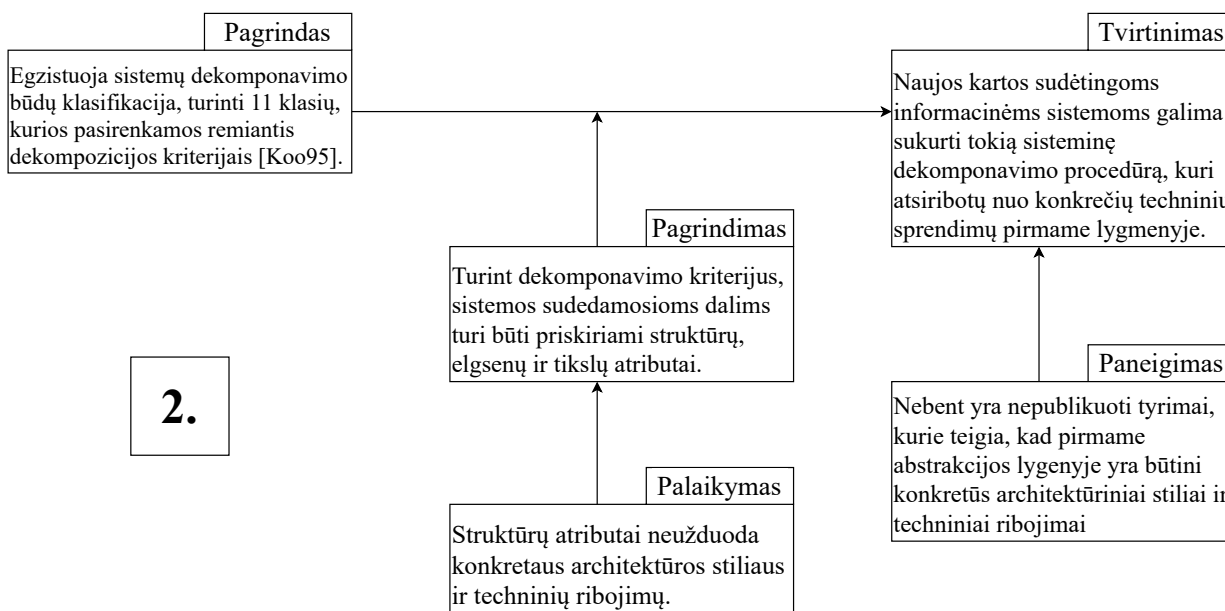
Kadangi pagrindinis baigiamojo darbo rezultatas yra teorinės išvados (kitai sakant, pasiūlytas dekomponavimo metodas), gautos deduktyviniais samprotavimais, tai rezultatų teisingumas gali būti pagrindžiamas analizuojant argumentų pagrįstumą ir argumentų prielaidų teisingumą. Argumentams konstruoti buvo taikomas Toulmino argumentavimo modelis [Tou03]. 31 pav. pateikta

Toulmino argumentavimo metodo schema, parodanti, koku principu darbe buvo konstruojami argumentai.

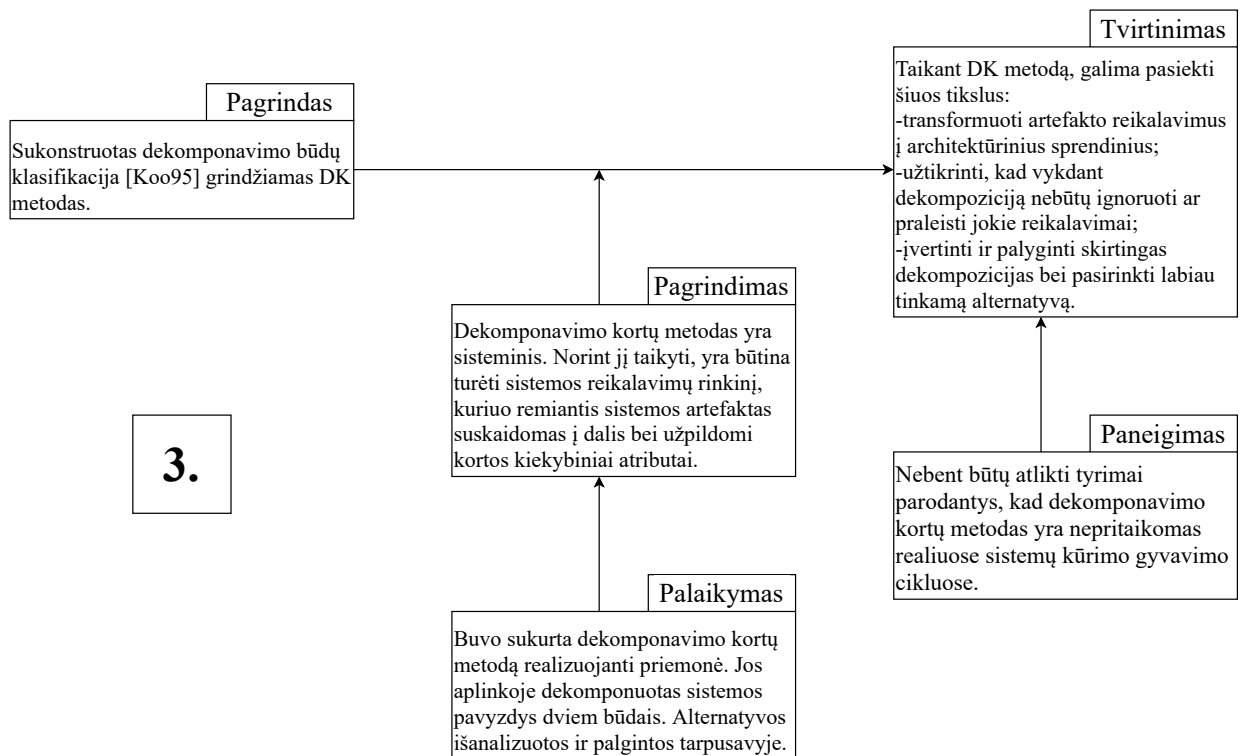
Šiame darbe sukonstruotų Toulmino argumentavimo schemų loginė grandinė pavaizduota eilės tvarka atitinkamai 32 pav., 33 pav. ir 34 pav.



32 pav. Pirmoji argumentavimo schema



33 pav. Antroji argumentavimo schema



34 pav. Trečioji argumentavimo schema

4.2. Metodo taikymo pavyzdys

Eksperimentą galima atlikti dar vienu būdu. Tai yra, pasirinkti esamą arba įsivaizduojamą sistemą ir pademonstruoti DK metodo veiksmingumą sukuriant sistemos dekompoziciją vienu arba keliais alternatyviais būdais. Šitaip būtų parodoma, jog naudojant DK metodą galima pasiekti jam išskeltus tikslus (žr. 3.3 poskyrį).

4.2.1. Dekomponuojamos sistemos pavyzdys

Eksperimentui atlikti autorius pasirinko sukurti įsivaizduojamą sistemą. Kadangi darbo tema apima naujos kartos sudėtingas informacines sistemas, buvo pasirinkta sukurti išmaniojo miesto (toliau – IM) sistemą, kuri atitiktų tiek sudėtingumo, tiek naujos kartos kriterijų.

Vykdant eksperimentą, dekompozicija buvo atlikta pirmame abstrakcijos lygmenyje, nes to pakanka metodo tinkamumo pagrindimui. Dekomponavimas antrame lygmenyje padidintų eksperimento apimtį ir būtų perteklinis. Sistemą taip pat pasirinkta dekomponuoti dviem skirtingais būdais, siekiant simuliuoti skirtingų architektų požiūrį į sistemų kūrimą ir tam, kad būtų pademonstruotas alternatyvų palyginimas.

Dekomponavimo veikla atlikta panaudojus autoriaus parengtą DK metodą realizuojantį įran-

kį⁹. Visi eksperimento metu gauti rezultatai patalpinti autoriaus „gitHub“ repozitorijoje¹⁰.

Norint pradėti IM sistemos dekompoziciją, reikia turėti savybių rinkinį. Kadangi abiem dekomponavimo būdais dekomponuojamas tas pats elementas tame pačiame lygmenyje, taikomas identiškas reikalavimų rinkinys, suskirstytas į 2.6 poskyryje aprašytas kategorijas. Svarbu pastebėti, kad savybių prioritetai ir tarpusavio sąveika taip pat išliks nepakitusi. Taigi, visa kairioji DK kortos dalis gali būti pakartotinai panaudota visose dekomponavimo alternatyvose (žr. 35 pav.). Savybių sąveikų matricos ir prioriteto reitingo atributai buvo užpildyti šio darbo autoriaus.

				Identifikatorius	Santykinis prioriteto reitingas	Prioriteto reitingas	Savybių sąrašas
				1	6%	4	E1: Eismo dalyviai turi važiuoti autonomiškai
				2	8%	5	E2: Eismo dalyviai turi gebėti bendrauti ir dalintis informacija su išore
				3	5%	3	E3: IM sistema valdo viešąjį transportą
		+		4	10%	6	E4: IM sistema stebi eismą ir pažeidimus
		+		5	10%	6	E5: IM sistema užtikrina šviesoforų ir dinaminių kelio ženklų reguliavimą
				6	16%	10	T1: Eismo dalyviai turi atitikti bent 4 žvaigždučių Euro NCAP saugumo standarto reitingą
				7	11%	7	T2: IM sistemos pelno šaltiniai turi būti reklamos ir prieiga prie duomenų
				8	8%	5	T3: IM sistema turi užtikrinti LR kelių eismo taisyklių laikymąsi
			+	9	15%	9	T4: IM sistema turi būti suderinama su policijos bei trečiųjų šalių sistemomis
				10	11%	7	T5: IM sistemos priežiūros kaina neturi viršyti 350 000 eurų per metus

35 pav. Užpildyta kairioji dekomponavimo kortos dalis

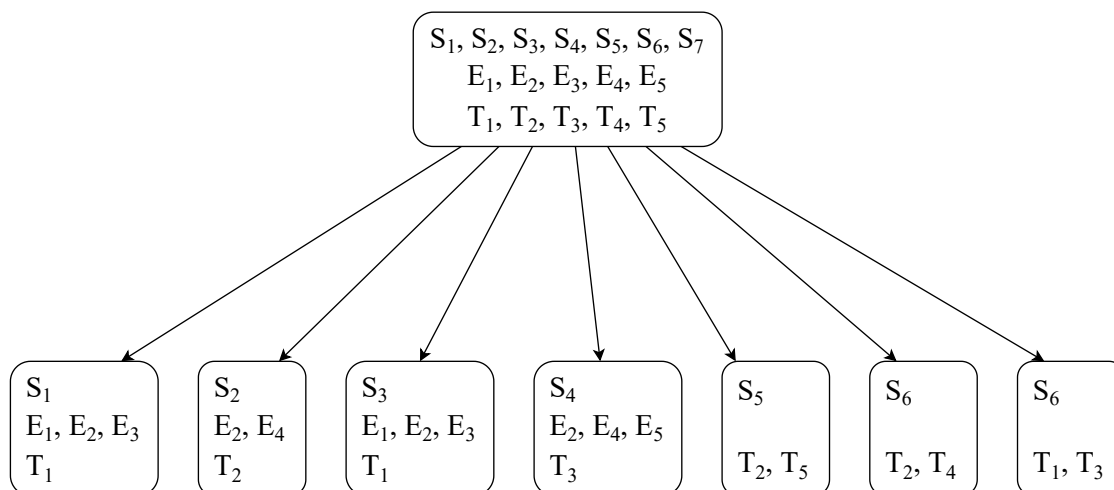
⁹Žr. <https://github.com/gedgeds/Decomposition-Cards-Method>; failas – „Dekomponavimo kortų įrankis.xlsx“; tikrinta 2021-05-09.

¹⁰Žr. <https://github.com/gedgeds/Decomposition-Cards-Method>; failas – „Pavyzdžio realizacija.xlsx“; tikrinta 2021-05-15.

4.2.2. Pirmoji dekompozicijos alternatyva

Šiame skirsnyje aprašyta IM sistemos dekompozicija pirmuoju būdu. Sistemos sudedamosios dalys išgautos skaidant sistemą į šias dalis:

- S1: Automobilio sistema;
- S2: Socialinio tinklo sistema;
- S3: Viešojo transporto valdymo sistema;
- S4: Eismo valdymo sistema;
- S5: Monetarinė sistema;
- S6: Trečiųjų šalių prieigos sistema;
- S7: Standartų užtikrinimo sistema.



36 pav. Pirmoji dekompozicija

Dekomponavus sistemą pirmuoju būdu, gauta grynoji struktūrų dekompozicijos klasė, kurios struktūrų, elgsenų ir tikslų paskirstymas pavaizduotas 36 pav. Pirmosios dekompozicijos alternatyvos viršutinės, centrinės ir dešinės kortos dalių užpildymas pateiktas 37 pav., o apatinė kortos dalis pateikta 38 pav. Sudedamųjų dalių atributų grupės kortos dalis įgalina analizuoti sudedamųjų dalių atributus, o kartu ir palyginti jas tarpusavyje. Išanalizavus šią atributų grupę, apie sudedamąsias dalis galima pasakyti tai:

- Aukščiausią svarbos reitingą turi „eismo valdymo sistema“, o žemiausią – „viešojo transporto valdymo sistema“. Rekomenduotina įgyvendinti ir skirti papildomą dėmesį visoms aukščiausią svarbos reitingą turinčioms sudedamosioms dalims, o nerealizavus žemiausio svarbos reitingo sistema neturėtų stipriai nukentėti.
- „Trečiųjų šalių prieigos sistema“ turi mažiausią atsakomybių verčių sumos reikšmę, todėl jos

įgyvendinimas ir palaikymas turėtų kainuoti mažiausiai pastangų.

- „Viešojo transporto valdymo sistema“ savybes realizuoja labiausiai paviršutiniškai. Taip yra dėl to, kad šios sudedamosios dalies atsakomybių verčių vidurkio reikšmė yra mažiausia.
- Daugiausiai ryšių turi „socialinio tinklo sistema“, todėl jos interfeisų sukūrimui ir priežiūrai reikės skirti daug pastangų. Tačiau mažiausiai ryšių turi dvi sudedamosios dalys: „automobilių aparatūra“ ir „trečiųjų šalių prieigos sistema“. Kadangi „automobilių aparatūra“ ryšių verčių sumos atributo reikšmė yra mažesnė, jos interfeisų kūrimas turėtų užimti mažiau pastangų.

Identifikatorius		Identifikatorius									
		1	2	3	4	5	6	7	Atsakingų dalių kiekis	Atsakingų dalių verčių suma	Atsakingų dalių verčių vidurkis
Savybių sąrašas		S1: Automobilio sistema	S2: Socialinio tinklo sistema	S3: Viešojo transporto valdymo sistema	S4: Eismo valdymo sistema	S5: Monetarinė sistema	S6: Trečiųjų šalių prieigos sistema	S7: Standartų užtikrinimo sistema			
1	E1: Eismo dalyviai turi važiuoti autonomiškai	●		▽					2	10	5,00
2	E2: Eismo dalyviai turi gebėti bendrauti ir dalintis informacija su išore	▽	●	▽	○				4	14	3,50
3	E3: IM sistema valdo viešąjį transportą	▽		●					2	10	5,00
4	E4: IM sistema stebi eismą ir pažeidimus		○		●				2	12	6,00
5	E5: IM sistema užtikrina šviesoforų ir dinaminių kelio ženklų reguliavimą				●				1	9	9,00
6	T1: Eismo dalyviai turi atitikti bent 4 žvaigždučių Euro NCAP saugumo standarto reitingą	●		▽				●	3	19	6,33
7	T2: IM sistemos pelno šaltiniai turi būti reklamos ir prieiga prie duomenų		○			●	▽		3	13	4,33
8	T3: IM sistema turi užtikrinti LR kelių eismo taisyklių laikymąsi				○			●	2	12	6,00
9	T4: IM sistema turi būti suderinama su policijos bei trečiųjų šalių sistemomis						●		1	9	9,00
10	T5: IM sistemos priežiūros kaina neturi viršyti 350 000 eurų per metus					●			1	9	9,00

37 pav. Užpildytos pirmosios dekompozicijos alternatyvos kortos dalys (1 dalis)

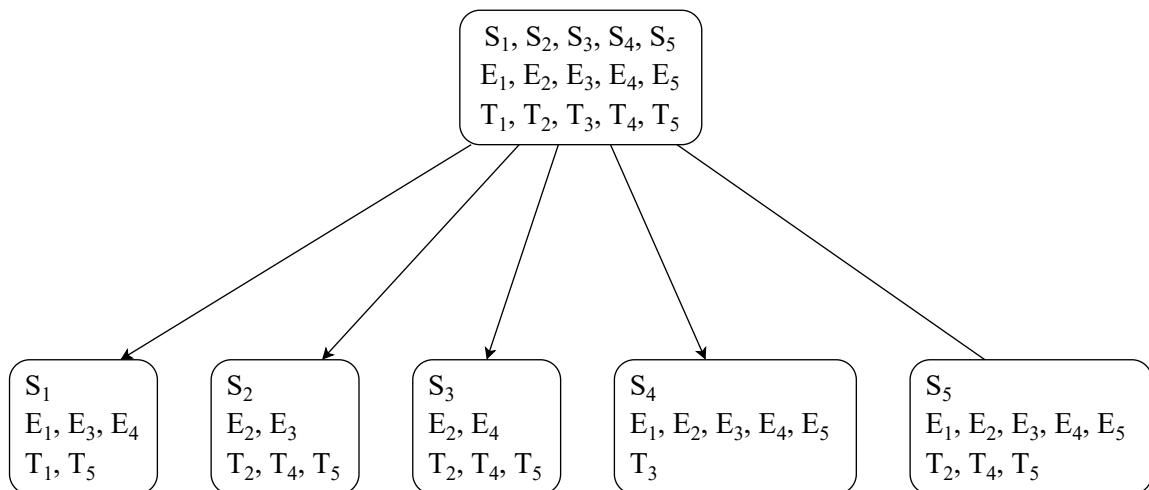
		Identifikatorius	1	2	3	4	5	6	7
Sudedamųjų dalių atributai	Svarbos reitingas	134	84	46	138	126	88	135	
	Santykinė svarba	18%	11%	6%	18%	17%	12%	18%	
	Atsakomybių kiekis	4	3	4	4	2	2	2	
	Atsakomybių verčių suma	20	15	12	24	18	10	18	
	Atsakomybių verčių vidurkis	5,00	5,00	3,00	6,00	9,00	5,00	9,00	
	Ryšių kiekis	2	4	3	3	3	2	3	
	Ryšių verčių suma	4	8	3	7	7	6	5	
	Ryšių verčių vidurkis	2,00	2,00	1,00	2,33	2,33	3,00	1,67	
Elemento atributai	Sudedamųjų dalių kiekis	7							
	Svarbos reitingų suma	751							
	Svarbos reitingų mediana	126							
	Santykinės svarbos rangas	0,12							
	Atsakomybių kiekis	21							
	Atsakomybių verčių suma	117							
	Atsakomybių verčių vidurkio rangas	6,00							
	Ryšių kiekis	10							
	Ryšių verčių suma	20							
	Ryšių kiekio rangas	2,00							

38 pav. Užpildytos pirmosios dekompozicijos alternatyvos kortos dalys (2 dalis)

4.2.3. Antroji dekompozicijos alternatyva

Šiame skirsnyje aprašyta IM sistemos dekompozicija antruoju būdu. Sistemos sudedamosios dalys išgautos skaidant sistemą į šias dalis:

- S1: Aparatūros dalis;
- S2: Duomenų dalis;
- S3: Komunikacijos dalis;
- S4: Skaičiavimų dalis;
- S5: Programų dalis.



39 pav. Antroji dekompozicija

Dekomponavus sistemą antruoju būdu, gauta grynoji struktūrų dekompozicijos klasė, kurios struktūrų, elgsenų ir tikslų paskirstymas pavaizduotas 39 pav. Antrosios dekompozicijos alternatyvos viršutinės, centrinės ir dešinės kortos dalių užpildymas pateiktas 40 pav., o apatinė kortos dalis pateikta 41 pav. Sudedamųjų dalių atributų grupės kortos dalis suteikia galimybę analizuoti sudedamųjų dalių atributus, o kartu ir palyginti jas tarpusavyje. Išanalizavus šią atributų grupę, apie sudedamąsias dalis galima pasakyti tai:

- Atsakomybių matricoje 7-tos ir 9-tos savybės sudedamųjų dalių atsakomybių pasiskirstymas sutampa, todėl vertėtų nuspręsti ar įmanoma šiuos du atskirus reikalavimus apjungti į vieną.
- „Programų dalis“ turi didžiulę svarbos reitingo reikšmę. Dėl to galima tvirtinti, kad šiai sudedamajai daliai reikia skirti papildomą dėmesį ir būtų rizikinga jos neįgyvendinti.
- „Programų dalis“ turi daugiausiai atsakomybių – 8, o tai reiškia, jog sudedamoji dalis yra atsakinga už 80% visų sistemos savybių. Tai yra milžiniška sudedamoji dalis, kurią yra rekomenduotina peržiūrėti. Ją įvertinus gali paaiškėti, kad esamajame lygmenyje jos negalima padalinti į smulkesnes dalis ir šitaip pasiekti optimalesnį atsakomybių paskirstymą.
- „Skaičiavimų dalis“ turi tik vieną ryšį ir šiai sudedamajai daliai reikės skirti mažiausiai resursų interfeisų kūrimui, o „programų dalis“ turi 4 ryšius ir didžiausią ryšių verčių sumos atributo reikšmę.
- Visų dekompozicijos sudedamųjų dalių ryšių verčių vidurkio atributo reikšmės yra didesnės už 2, o tai reiškia, kad dekompozicijoje vyrauja stiprūs tarpusavio ryšiai. Todėl galima daryti prielaidą, kad šioje dekompozicijoje reikės skirti daug pastangų interfeisų kūrimui ir priežiūrai.

Identifikatorius		Identifikatorius							
		1	2	3	4	5	Atsakingų dalių kiekis	Atsakingų dalių verčių suma	Atsakingų dalių verčių vidurkis
Savybių sąrašas	Sudedamosios dalys	S1: Aparatūros dalis	S2: Duomenų dalis	S3: Komunikacijos dalis	S4: Skaičiavimų dalis	S5: Programų dalis			
		1	E1: Eismo dalyviai turi važiuoti autonomiškai	▽			●	▽	3
2	E2: Eismo dalyviai turi gebėti bendrauti ir dalintis informacija su išore		○	●	▽	●	4	22	5,50
3	E3: IM sistema valdo viešąjį transportą	▽	▽		○	●	4	14	3,50
4	E4: IM sistema stebi eismą ir pažeidimus	●		○	○	●	4	24	6,00
5	E5: IM sistema užtikrina šviesoforų ir dinaminių kelio ženklų reguliavimą				●	○	2	12	6,00
6	T1: Eismo dalyviai turi atitikti bent 4 žvaigždučių Euro NCAP saugumo standarto reitingą	●					1	9	9,00
7	T2: IM sistemos pelno šaltiniai turi būti reklamos ir prieiga prie duomenų		●	○		○	3	15	5,00
8	T3: IM sistema turi užtikrinti LR kelių eismo taisyklių laikymąsi				●		1	9	9,00
9	T4: IM sistema turi būti suderinama su policijos bei trečiųjų šalių sistemomis		●	○		○	3	15	5,00
10	T5: IM sistemos priežiūros kaina neturi viršyti 350 000 euru per metus	●	○	▽		●	4	22	5,50

40 pav. Užpildytos antrosios dekompozicijos alternatyvos kortos dalys (1 dalis)

		Identifikatorius	1	2	3	4	5
Sudedamųjų dalių atributai	Svarbos reitingas	214	183	118	167	259	
	Santykinė svarba	23%	19%	13%	18%	28%	
	Atsakomybių kiekis	5	5	5	6	8	
	Atsakomybių verčių suma	29	25	19	34	46	
	Atsakomybių verčių vidurkis	5,80	5,00	3,80	5,67	5,75	
	Ryšių kiekis	3	3	3	1	4	
	Ryšių verčių suma	7	7	7	3	10	
	Ryšių verčių vidurkis	2,33	2,33	2,33	3,00	2,50	
Elemento atributai	Sudedamųjų dalių kiekis	5					
	Svarbos reitingų suma	941					
	Svarbos reitingų mediana	183					
	Santykinės svarbos rangas	0,15					
	Atsakomybių kiekis	29					
	Atsakomybių verčių suma	153					
	Atsakomybių verčių vidurkio rangas	2,00					
	Ryšių kiekis	7					
	Ryšių verčių suma	17					
	Ryšių kiekio rangas	3,00					

41 pav. Užpildytos antrosios dekompozicijos alternatyvos kortos dalys (2 dalis)

4.2.4. Dekompozicijos alternatyvų palyginimas

Šiame skirsnyje analizuojamas abiejų IM sistemos dekompozicijų palyginimas. Lyginimo metu bus gretinami savybių atributų ir elemento atributų aibės.

42 pav. pateiktas savybių atributų gretinimas, įgalinantis analizuoti konkrečias savybes. Iš užpildytų duomenų darbo autorius pateikia šiuos pastebėjimus ir išvagas:

- Abiejose alternatyvose nėra nulinės atsakingų dalių kiekio atributo reikšmės. Tai reiškia, kad nei viena savybė nebuvo praleista dekompozicijos metu. Tiesa, to neužtenka daryti apibendrinimą, kad dekompozicijose nėra paviršutiniškai realizuojamų savybių.
- Pirmoji alternatyva turi tik vieną savybę su 4-iomis už ją atsakingomis dalimis, o antroji

alternatyva net 4-ias tokias savybes. Be to, antroji alternatyva turi tik 5-ias sudedamąsias dalis, o tai reiškia, kad kiekvieną iš 4-ių jau minėtų savybių realizuoja 4-ios iš 5-ių sudedamųjų dalių. Jeigu šis pavyzdys būtų analizuojamas ne pirmame abstrakcijos lygmenyje, būtų galima rekomenduoti perkelti savybes į aukštesnį lygmenį, nes esamajame jos yra nelokalizuojamos. Tačiau dekompozicijos pavyzdys yra pirmame lygmenyje, todėl galima teigti, jog antroji dekompozicijos alternatyva sudarys sunkesnes sąlygas atlikti savybių įgyvendinimą.

- Atsakingų dalių verčių sumos atributas 8-iose iš 10-ies savybių yra aukštesnis antroje alternatyvoje negu pirmoje. Be to, 3-ose iš 10-ies savybėse atributas viršija reikšmę 20, kai pirmosios alternatyvos visų savybių reikšmės yra mažesnės už 20. Su tokiu atributo reikšmių paskirstymu yra susijęs prieš tai esantis punktas, kuriame antroji alternatyva turi tankesnę savybes realizuojančių dalių pasiskirstymą. Aukštos savybių atsakingų dalių verčių sumos atributo reikšmės indikuoja, kad antroje alternatyvoje bus sunkiau realizuoti savybių užtikrinimą ir bet kokie pokyčiai reikalaus daugiau pastangų negu pirmojoje alternatyvoje.
- Atsakingų dalių verčių vidurkio atributo reikšmės abiejose alternatyvose yra panašūs ir neturi labai žemų reikšmių, o tai reiškia, kad nėra paviršutiniškai realizuojamų savybių.
- Antrosios savybės „E2: Eismo dalyviai turi gebėti bendrauti ir dalintis informacija su išore“ atsakingų dalių kiekio atributo reikšmės sutampa abiejose alternatyvose. Tačiau pagrindinis skirtumas pasimato atsakingų dalių verčių sumos atributo reikšmėje. Pirmoje alternatyvoje atributo reikšmė yra 14, o antroje alternatyvoje – 22. Taigi, antroje alternatyvoje antrojo reikalavimo įgyvendinimui reikės daugiau pastangų.
- Aukščiausio prioriteto savybė „T1: Eismo dalyviai turi atitikti bent 4 žvaigždučių Euro NCAP saugumo standarto reitingą“ pirmoje alternatyvoje pasižymi 3 už ją atsakingomis sudedamosiomis dalimis, o atsakomybių verčių suma yra 19. Tačiau antroje alternatyvoje už savybę yra atsakinga tik 1 dalis ir jos atsakomybės stiprumo reikšmė yra 9. Šiuo konkrečiu atveju antroji alternatyva yra pranašesnė, nes savybė yra lokalizuota, t. y. įgyvendinama tik vienoje sudedamojoje dalyje, o tai reikalaus mažiau pastangų savybės užtikrinimui.

Identifikatorius	Santykinis prioriteto reitingas	Prioriteto reitingas	Savybių sąrašas	Alternatyva #1			Alternatyva #2		
				Atsakingų dalių kiekis	Atsakingų dalių verčių suma	Atsakingų dalių verčių vidurkis	Atsakingų dalių kiekis	Atsakingų dalių verčių suma	Atsakingų dalių verčių vidurkis
1	6%	4	E1: Eismo dalyviai turi važiuoti autonomiškai	2	10	5,00	3	11	3,67
2	8%	5	E2: Eismo dalyviai turi gebėti bendrauti ir dalintis informacija su išore	4	14	3,50	4	22	5,50
3	5%	3	E3: IM sistema valdo viešąjį transportą	2	10	5,00	4	14	3,50
4	10%	6	E4: IM sistema stebi eismą ir pažeidimus	2	12	6,00	4	24	6,00
5	10%	6	E5: IM sistema užtikrina šviesoforų ir dinaminių kelio ženklų reguliavimą	1	9	9,00	2	12	6,00
6	16%	10	T1: Eismo dalyviai turi atitikti bent 4 žvaigždučių Euro NCAP saugumo standarto reitingą	3	19	6,33	1	9	9,00
7	11%	7	T2: IM sistemos pelno šaltiniai turi būti reklamos ir prieiga prie duomenų	3	13	4,33	3	15	5,00
8	8%	5	T3: IM sistema turi užtikrinti LR kelių eismo taisyklių laikymąsį	2	12	6,00	1	9	9,00
9	15%	9	T4: IM sistema turi būti suderinama su policijos bei trečiųjų šalių sistemomis	1	9	9,00	3	15	5,00
10	11%	7	T5: IM sistemos priežiūros kaina neturi viršyti 350 000 eurų per metus	1	9	9,00	4	22	5,50

42 pav. Dekomponavimo kortų alternatyvų savybių atributų palyginimas

43 pav. pateiktas elemento atributų gretinimas, įgalinantis analizuoti dekomponuojamą elementą. Iš užpildytų duomenų darbo autorius pateikia šiuos pastebėjimus ir išvalgas:

- Pirmoje alternatyvoje yra 7 sudedamosios dalys, o antroje – 5. Tikėtina, kad ryšių kiekis bus didesnis pirmojoje alternatyvoje, nes visų įmanomų ryšių gali susidaryti 21, kai antroje

alternatyvoje įmanomi tik 10 ryšių. Jeigu visų kiti elemento atributų reikšmės sutaptų, būtų galima daryti prielaidą, kad pirmoji dekompozicija yra aukštesnio sudėtingumo dėl didesnio sudedamųjų dalių kiekio.

- Svarbos reitingų sumos atributas gerokai didesnis antroje dekompozicijoje – atitinkamai 941 ir 751. Dar didesnis skirtumas pasireiškia svarbos reitingų medianos atributo reikšmėse – 183 (antroji dekompozicija) ir 126 (pirmoji dekompozicija). Šie skirtumai reiškia tai, kad antrosios dekompozicijos sudedamosios dalys turi didesnę įtaką visai sistemai. Tai yra, nusprendus neįgyvendinti vienos sudedamosios dalies antroje alternatyvoje sistema vidutiniškai patirs didesnę žalą, negu pirmoje alternatyvoje. Didesnė žala susidarys dėl to, kad svarbos reitingų sumos atributo dydis priklauso nuo visų sudedamųjų dalių turimų atsakomybių už savybes, o tai reiškia, kad atsisakius tam tikrų sudedamųjų dalių padidėja rizika neįgyvendinti sistemos savybių. Mediana nurodo vidurinę sudedamosios dalies svarbą elemente.
- Pirmosios alternatyvos sudedamosios dalys yra šiek tiek labiau subalansuotos, nes skirtumas tarp didžiausių ir mažiausių svarbą turinčių dalių yra mažesnis.
- Nors antroje dekompozicijoje sudedamųjų dalių yra mažiau negu pirmoje, atsakomybių kiekis yra didesnis antroje alternatyvoje – atitinkamai 29 ir 21. Panaši proporcija išlieka ir atsakomybių verčių sumos atributo reikšmėse – 153 (antroji dekompozicija) ir 117 (pirmoji dekompozicija).
- Tarp abiejų dekompozicijų pasireiškia didžiulis skirtumas atsakomybių verčių vidurkio rango atributo reikšmėse. Pirmosios dekompozicijos rango reikšmė yra lygi 6, o tai reiškia, kad egzistuoja didelis skirtumas tarp mažiausių ir didžiausių atsakomybių verčių sumos skirtumą turinčių sudedamųjų dalių. Ir priešingai, antrosios dekompozicijos rango reikšmė yra 2. Tai reiškia, kad atsakomybių paskirstymas sudedamosioms dalims yra labiau subalansuotas: arba dauguma dalių turi stiprias atsakomybes, arba dauguma dalių turi silpnas atsakomybes.
- Ryšių kiekio ir ryšių verčių suma didesnė pirmoje alternatyvoje, o tai reiškia, kad reikės sukurti daugiau interfeisų tarp ryšius turinčių dalių ir komunikacijos srautai bus intensyvesni negu antroje dekompozicijoje, o tai neigiamai paveiks sistemos sudėtingumą. Tačiau antroje alternatyvoje ryšių kiekis artėja prie dydžio $\frac{n^2-n}{2}$ (čia n – sudedamųjų dalių kiekis), o tai indikuoja apie nekokybišką dekompoziciją, kurioje faktiškai egzistuoja tiek ryšių, kiek yra visų teoriškai galimų ryšių.
- Ryšių kiekio rangas yra didesnis antroje alternatyvoje, o tai reiškia, kad nors pirmoji dekompozicija ir turi daugiau tarpusavio ryšių, sudedamosios dalys yra labiau subalansuotos, t. y. dekompozicijoje yra mažiau dalių, kurios pasižymi labai dideliu ryšių kiekiu.

Elemento atributai	Sudedamųjų dalių kiekis	7	5
	Svarbos reitingų suma	751	941
	Svarbos reitingų mediana	126	183
	Santykinės svarbos rangas	0,12	0,15
	Atsakomybių kiekis	21	29
	Atsakomybių verčių suma	117	153
	Atsakomybių verčių vidurkio rangas	6,00	2,00
	Ryšių kiekis	10	7
	Ryšių verčių suma	20	17
	Ryšių kiekio rangas	2,00	3,00
		Alternatyva #1	Alternatyva #2

43 pav. Dekomponavimo kortų alternatyvų elemento atributų palyginimas

Autoriaus vertinimu, norint dekomponuoti IM sistemą tokiu būdu, kuris sumažintų jos sudėtingumą pirmame lygmenyje, reikia pasirinkti pirmąją dekompozicijos alternatyvą, nes:

- Antroje alternatyvoje gali būti sunkiau realizuoti tam tikras savybes;
- Svarbos reitingų suma didesnė antroje dekompozicijoje;
- Nors sudedamųjų dalių yra daugiau pirmoje alternatyvoje, atsakomybių kiekis ir atsakomybių verčių suma yra gerokai didesnė antroje alternatyvoje;
- Pirmoji dekompozicija turi šiek tiek didesnę ryšių kiekį ir ryšių verčių sumą, bet antrosios alternatyvos dauguma ryšių yra stiprūs;
- Gali reikėti peržiūrėti antrąją alternatyvą ir atlikti papildomus pakeitimus.

Apibendrinant galime teigti, kad dekomponavimo kortų kiekybinių reikšmių palyginimas gali suteikti naudingą duomenų apie sistemos savybes ir dekomponuojamus elementus. Turint iš anksto užsibrėžtus atrankos kriterijus, dekompozicijų palyginimas gretinant kortų atributus tarpusavyje gali padėti apsispręsti kurią alternatyvą pasirinkti.

Rezultatai ir išvados

Baigiamojo darbo **teoriniai rezultatai** yra šie:

1. Išnagrinėtos ir apibrėžtos naujos kartos informacinių sistemų savybės, turinčios tiesioginę įtaką jų sudėtingumui.
2. Pasiūlyta, kaip spręsti didelių sudėtingų sistemų dekomponavimo problemą, kai turi būti tenkinami šie reikalavimai:
 - dekomponavimas turi užtikrinti mažesnę sprendinio sudėtingumą, palyginus su nedekomponuota sistema;
 - sudarant dekompoziciją turi būti atsižvelgiama į visus reikalavimus ir dekomponavimo kriterijus (t.y., užtikrinti išsamumą);
 - sprendinys neturi priklausyti nuo nepakankamos sistemos architekto kvalifikacijos.
3. Pasiūlyti dekompozicijų sudėtingumo matavimo atributai, siekiant užtikrinti galimybę palyginti alternatyvius sprendinius.

Taip pat buvo gauti šie **praktiniai rezultatai**:

1. Sukurtas dekomponavimo kortų metodą realizuojantis praktinis įrankis. Šis įrankis įgalina tirti galimas dekompozicijas, be to, palengvina sistemos kūrėjų darbą automatizuojant didelę dalį kortos atributų skaičiavimų.
2. sudarytos dvi sistemos dekompozicijos skirtingais būdais taikant dekomponavimo kortų metodą realizuojančią priemonę. Taip pat atliktas abiejų dekompozicijos alternatyvų įvertinimas ir argumentuotas palyginimas.

Baigiamojo darbo „Naujos kartos sudėtingų informacinių sistemų dekomponavimas“ **išvados** yra šios:

1. Atlikus literatūros analizę nustatyta, kad kuriant sistemas pagrindinis dėmesys yra skiriamas ir yra taikomi tie dekomponavimo būdai, kurie jau pirmajame sistemos dekomponavimo lygmenyje užduoda konkrečią architektūros specifiką ir techninius ribojimus.
2. Norint užtikrinti sistemine ir kokybišką naujos kartos sudėtingų informacinių sistemų dekompoziciją, reikia taikyti dekomponavimo kortų metodą:
 - metodas įgyvendina ne konkretaus architektūros stiliaus, o hierarchinį dekomponavimo būdą, kur aukštesniame lygmenyje identifikuojami sudedamieji sistemos elementai, o

žemesniame lygmenyje nustatomos tų elementų dalys.

- metodas susieja reikalavimų išsiaiškinimo ir projektavimo veiklas, kai artefaktų (t.y., sistemos ar jos dalių) savybes ir bazinius jų atributus apibrėžia reikalavimų inžinierius, o jais remiantis architektas dekomponuoja artefaktą į sudedamąsias dalis.
- naudojant šį metodą, kelių skirtingų dekomponavimo būdų analizė ir palyginimas turi būti vykdomas gretinant juos nusakančių atributų reikšmes tarpusavyje (reikalavimų aspektu – savybių atributų grupė kiekvienam reikalavimui individualiai; sudedamųjų dalių aspektu – iš visų dekomponuotame elemente išvestų sudedamųjų dalių sintezuotų elemento atributų grupė).

3. Atlikus atvejo analizę nustatyta, kad metodą galima taikyti praktinei dekompozicijos veiklai vykdyti. Išanalizavus konkrečias dekompozicijas įvertinami jų sudėtingumui įtaką darantys kiekybiniai atributai. Dekomponavimo kortų pagalba pagrįstos įžvalgos įgalina pasirinkti tinkamą sistemos dekompozicijos alternatyvą.
4. Dekomponavimo kortų metodą realizuojanti priemonė sudaro prielaidas tolimesniems eksperimentiniams metodo tyrimams vykdyti. Vertinant šią priemonę nustatyta, kad be dekomponavimo būdo tinkamumo vertinimui reikalingų duomenų pateikimo ir kiekybinių įverčių skaičiavimo, galima automatizuoti ir kitas metodu numatytas veiklas (kortos dalių pildymo eiliškumo užtikrinimą, tendencijų atpažinimą, kūrėjo pateikiamų duomenų teisingumo kontrolę, dekomponavimo alternatyvų įverčių importavimą, sudedamųjų dalių ir savybių kiekio keitimą).

Sąvokų apibrėžimai

Atributas – DK metodo kortos dalis, atsakinga už konkretų dekompozicijos aspekto kiekybinę įvertį.

Automatinis rodiklis – DK metodo kortos atributas, kurio reikšmės gaunamos automatiškai, pavyzdžiui identifikatoriai arba importuojami atributai.

Bazinis rodiklis – DK metodo kortos atributas, kurio reikšmę nurodo metodą naudojantis specialistas.

Dekompozicijos lygmuo – sistemos architektūrinis abstrakcijos lygmuo, kuriame yra atliekama artefakto dekompozicija

Išvestinis rodiklis – DK metodo kortos atributas, kurio reikšmei gauti yra būtinos kitų atributų reikšmės ir kuris yra apskaičiuojant naudojant konkrečią formulę.

Nagrinėjamas elementas – artefaktas, kurio dekompozicija yra atliekama DK metodo kortos kontekste.

Savybė – nagrinėjamam elementui išskelti reikalavimai.

Sudedamoji dalis – dekomponavimo metu gauta smulkesnė sudaromoji nagrinėjamo elemento dalis.

Tendencija – DK metodo kortos atributo reikšmių visumos atitikimas iš anksto apibrėžtiems požymiams.

Santrumpos

BPM – Verslo procesų valdymas (angl. *business process management*).

DBVS – Duomenų bazių valdymo sistema.

DK – Dekomponavimo korta.

ERP – Verslo valdymo sistema (angl. *enterprise resource planning*).

HMI – Žmogaus ir mašinos sąveika (angl. *human-machine interaction*).

IBM – JAV kompiuterių gamintoja, programinės ir techninės įrangos prekybos bendrovė, teikianti infrastruktūros ir konsultavimo paslaugas (angl. *International Business Machines corporation*).

IEC – Tarptautinė elektrotechnikos komisija, atsakinga už elektrinės įrangos standartizavimą (angl. *International Electrotechnical Commission*).

IM – Išmaniojo miesto sistema.

IS – Informacinė sistema.

ISO – Tarptautinė standartizacijos organizacija, kurianti tarptautinius standartus (angl. *International Organization for Standardization*).

IT – Informacinės technologijos.

KFS – Kokybės funkcijų sklaida (angl. *quality function deployment*).

MRP – Išteklių reikalavimų planavimas (angl. *material requirements planning*).

NFR – Nefunkcinių reikalavimų karkasas (angl. *Non-functional requirements framework*).

VLKK – Valstybinė lietuvių kalbos komisija.

Literatūra

- [AGH⁺03] E. Ammenwerth, S. Gräber, G. Herrmann, T. Bürkle & J. König. Evaluation of health information systems—problems and challenges. *International journal of medical informatics*, 71(2-3):125–135, 2003. [žiūrėta 2020-03-22]. Prieiga per internetą: <<http://schwabweb.de/Publikationen/z19.pdf>>.
- [Art18] Brian W. Arthur. *The economy as an evolving complex system II*. CRC Press, 2018. [žiūrėta 2020-03-07]. Prieiga per internetą: <<https://www.econ.tuwien.ac.at/lva/compeco.se/artikel>>.
- [BCG⁺08] P. Boxer, D. Carney, S. Garcia, L. Brownsword, W. Anderson, P. Kirwan, D. Smith & J. Morley. SoS Navigator 2.0: A Context-Based Approach to System-of-Systems Challenges. Tech. atask., Carnegie-Mellon University, Pittsburgh PA, Software Engineering Institute, 2008. [žiūrėta 2020-03-07]. Prieiga per internetą: <<https://apps.dtic.mil/dtic/tr/fulltext/u2/a489051.pdf>>.
- [BD08] Monica F. Bugallo & Petar M. Djuric. Complex systems and particle filtering. *2008 42nd Asilomar Conference on Signals, Systems and Computers*, p. 1183–1187. IEEE, 2008. [žiūrėta 2020-03-08]. Prieiga per internetą: <<https://ieeexplore.ieee.org/abstract/document/5074602>>.
- [Ber03] S. Bertelsen. Construction as a complex system. *Proceedings for the 11th annual conference of the International Group for Lean Construction*, p. 11–23, 2003. [žiūrėta 2020-03-07]. Prieiga per internetą: <https://www.researchgate.net/profile/Sven_Bertelsen/publication/236841812>.
- [BPV20] Lisa H. Broska, W. Poganietz & S. Vögele. Extreme events defined — a conceptual discussion applying a complex systems approach. *Futures*, 115:102490, 2020. [žiūrėta 2020-03-08]. Prieiga per internetą: <<https://www.sciencedirect.com/science/article/pii/S0016328719303520>>.
- [Bro01] Tyson R. Browning. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering management*, 48(3):292–306, 2001. [žiūrėta 2020-02-10]. Prieiga per internetą: <<https://www.researchgate.net/publication/3076682>>.

- [CHL⁺11] N. Chiriac, K. Hölttä-Otto, D. Lysy & Eun S. Suh. Three approaches to complex system decomposition. *DSM 2011: Proceedings of the 13th International DSM Conference*, 2011. [žiūrēta 2020-02-10]. Prieiga per internetą: <<https://www.designsociety.org/download-publication/30820>>.
- [CKL12] P. Colombo, F. Khendek & L. Lavazza. Bridging the gap between requirements and design: an approach based on problem frames and sysml. *Journal of Systems and Software*, 85(3):717–745, 2012. [žiūrēta 2020-06-13]. Prieiga per internetą: <<https://www.sciencedirect.com/science/article/pii/S0164121211002561>>.
- [CT19] S. Chaabane & D. Trentesaux. Coping with disruptions in complex systems: a framework. *IFAC-PapersOnLine*, 52(13):2413–2418, 2019. [žiūrēta 2020-03-08]. Prieiga per internetą: <<https://www.sciencedirect.com/science/article/pii/S2405896319315551>>.
- [CW02] Lai K. Chan & Ming L. Wu. Quality function deployment: a literature review. *European journal of operational research*, 143(3):463–497, 2002. [žiūrēta 2021-01-16]. Prieiga per internetą: <<https://www.sciencedirect.com/science/article/abs/pii/S0377221702001789>>.
- [DFH16] F. Dalpiaz, X. Franch & J. Horkoff. Istar 2.0 language guide. *arXiv preprint arXiv:1605.07767*, 2016. [žiūrēta 2021-01-17]. Prieiga per internetą: <<https://arxiv.org/pdf/1605.07767.pdf>>.
- [DLK11] Tung M. Dao, H. Lee & Kyo C. Kang. Problem frames-based approach to achieving quality attributes in software product line engineering. *2011 15th International Software Product Line Conference*, p. 175–180. IEEE, 2011. [žiūrēta 2020-06-13]. Prieiga per internetą: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6030059>>.
- [HC88] John R. Hauser & D. Clausing. The house of quality. *Harvard Business Review*, 1988. [žiūrēta 2021-01-16]. Prieiga per internetą: <<http://blogs.ubc.ca/nvdteamb/files/2013/10/7-The-House-of-Quality.pdf>>.
- [HG99] James D. Herbsleb & Rebecca E. Grinter. Architectures, coordination, and distance: conway’s law and beyond. *IEEE software*, 16(5):63–70, 1999. [žiūrēta 2020-02-10]. Prieiga per internetą: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=795103>>.

- [HHS06] D. Hatebur, M. Heisel & H. Schmidt. Security engineering using problem frames. *International Conference on Emerging Trends in Information and Communication Security*, p. 238–253. Springer, 2006. [žiūrėta 2020-06-13]. Prieiga per internetą: <<https://www.researchgate.net/publication/221536560>>.
- [HPD08] J. Holmgren, Jan A. Persson & P. Davidsson. Agent based decomposition of optimization problems. *First International Workshop on Optimization in Multi-Agent Systems*, 2008. [žiūrėta 2020-06-13]. Prieiga per internetą: <<http://www.diva-portal.org/smash/get/diva2:836280>>.
- [IEC11] ISO IEC. ISO/IEC 25010:2011, systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models, 2011. [žiūrėta 2021-01-10]. Prieiga per internetą: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>>.
- [YM09] A. Yrjönen & J. Merilinna. Extending the nfr framework with measurable nonfunctional requirements. *NFPinDSML MoDELS*, 2009. [žiūrėta 2021-01-17]. Prieiga per internetą: <<https://www.researchgate.net/publication/38289214>>.
- [Koo95] P. Koopman. A taxonomy of decomposition strategies based on structures, behaviors, and goals. *Design Theory & Methodology Conference*, 1995. [žiūrėta 2020-06-13]. Prieiga per internetą: <<http://users.ece.cmu.edu/~koopman/decomp/decomp.pdf>>.
- [Kra21] G. Krasauskas. Dekomponavimo kortų metodas sudėtingoms informacinėms sistemoms dekomponuoti. *Vilnius University Open Series*:44–52, 2021. [žiūrėta 2021-05-14]. Prieiga per internetą: <<https://www.zurnalai.vu.lt/open-series/article/view/24025/23329>>.
- [KT11a] H. Komoto & T. Tomiyama. A theory of decomposition in system architecting. *DS 68-2: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 2: Design Theory and Research Methodology, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011*, 2011. [žiūrėta 2020-06-13]. Prieiga per internetą: <<https://www.designsociety.org/download-publication/30484>>.

- [KT11b] H. Komoto & T. Tomiyama. Multi-disciplinary system decomposition of complex mechatronics systems. *CIRP annals*, 60(1):191–194, 2011.
[žiūrēta 2020-06-13]. Prieiga per internetą:
<<https://www.sciencedirect.com/science/article/abs/pii/S000785061100103X>>.
- [KT12] H. Komoto & T. Tomiyama. A framework for computer-aided conceptual design and its application to system architecting of mechatronics products. *Computer-Aided Design*, 44(10):931–946, 2012. [žiūrēta 2020-06-13]. Prieiga per internetą:
<<https://www.sciencedirect.com/science/article/abs/pii/S0010448512000401>>.
- [LBJ⁺04] R. Laney, L. Barroca, M. Jackson & B. Nuseibeh. Composing requirements using problem frames. *Proceedings. 12th IEEE International Requirements Engineering Conference, 2004*. P. 122–131. IEEE, 2004.
[žiūrēta 2020-06-13]. Prieiga per internetą:
<<https://ieeexplore.ieee.org/document/1335670>>.
- [LLW13] J. Ladyman, J. Lambert & K. Wiesner. What is a complex system? *European Journal for Philosophy of Science*, 3(1):33–67, 2013.
[žiūrēta 2020-02-10]. Prieiga per internetą:
<<http://philsci-archive.pitt.edu/9044/4/LLWultimate.pdf>>.
- [Löw19] J. Löwy. *Righting Software*. Addison-Wesley Professional, 2019.
[žiūrēta 2020-02-10]. Prieiga per internetą:
<<https://www.informit.com/articles/article.aspx?p=2995357>>.
- [MB17] Glenn H. Mazur & QFD Red Belt. Beyond iso 16355: qfd for a digital world, 2017.
[žiūrēta 2021-01-16]. Prieiga per internetą:
<<http://www.mazur.net/works/Beyond-ISO-16355>>.
- [MR13] E. Merelli & M. Rasetti. Non locality, topology, formal languages: new global tools to handle large data sets. *Procedia Computer Science*, 18:90–99, 2013.
[žiūrēta 2020-03-08]. Prieiga per internetą:
<<https://www.sciencedirect.com/science/article/pii/S1877050913003153>>.
- [NFT16] F. Niederman, Thomas W. Ferratt & Eileen M. Trauth. On the co-evolution of information technology and information systems personnel. *ACM SIGMIS Database: the Database for Advances in Information Systems*, 47(1):29–50, 2016.

- [žiūrėta 2020-02-10]. Prieiga per internetą:
<<https://ecommons.udayton.edu/cgi/1082>>.
- [Pen11] B. Penzenstadler. *DeSyRe-decomposition of systems and their requirements*. Disertacija, Technische Universität München, 2011.
- [žiūrėta 2021-01-10]. Prieiga per internetą:
<<https://mediatum.ub.tum.de/doc/999357/file.pdf>>.
- [PZJ⁺16] H. Panetto, M. Zdravkovic, R. Jardim-Goncalves, D. Romero, J. Cecil & I. Mezgár. New perspectives for the future interoperable enterprise systems. *Computers in Industry*, 79:47–63, 2016. [žiūrėta 2020-06-13]. Prieiga per internetą:
<<https://hal.archives-ouvertes.fr/hal-01142747/document>>.
- [RHJ⁺04] L. Rapanotti, Jon G. Hall, M. Jackson & B. Nuseibeh. Architecture-driven problem decomposition. *Proceedings. 12th IEEE International Requirements Engineering Conference, 2004*. P. 80–89. IEEE, 2004. [žiūrėta 2020-06-13]. Prieiga per internetą:
<<https://ieeexplore.ieee.org/document/1335666>>.
- [RMC98] Jack B. Revelle, John W. Moran & Charles A. Cox. *The QFD handbook*. John Wiley & Sons, 1998. [žiūrėta 2021-01-16]. Prieiga per internetą:
<<https://books.google.nl/books?id=ngbWPXF118C&oi>>.
- [Rup10] Nayan B. Ruparelia. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3):8–13, 2010. [žiūrėta 2020-05-10]. Prieiga per internetą:
<<https://dl.acm.org/doi/pdf/10.1145/1764810.1764814>>.
- [RV16] D. Romero & F. Vernadat. Enterprise information systems state of the art: past, present and future trends. *Computers in Industry*, 79:3–13, 2016.
- [žiūrėta 2020-02-10]. Prieiga per internetą:
<https://www.researchgate.net/profile/David_Romero2/publication/297816747>.
- [SCH15] Eun S. Suh, N. Chiriac & K. Hölttä-Otto. Seeing complex system through different lenses: impact of decomposition perspective on system architecture analysis. *Systems Engineering*, 18(3):229–240, 2015. [žiūrėta 2020-06-13]. Prieiga per internetą:
<<https://onlinelibrary.wiley.com/doi/pdf/10.1002/sys.21294>>.
- [SLL⁺18] A. Shahbazian, Youn K. Lee, D. Le, Y. Brun & N. Medvidovic. Recovering architectural design decisions. *2018 IEEE International Conference on Software Architecture*

- (ICSA), p. 95–9509. IEEE, 2018. [žiūrėta 2020-02-10]. Prieiga per internetą:
<<https://arxiv.org/pdf/1704.04798.pdf>>.
- [Tag11] N. Tagoug. Information system decomposition quality. *Journal of Digital Information Management*, 9(3), 2011. [žiūrėta 2020-06-13]. Prieiga per internetą:
<<http://dline.info/fpaper/jdim/v9i3/2.pdf>>.
- [TYW99] C. Temponi, J. Yen & Amos W. Tiao. House of quality: a fuzzy logic-based requirements analysis. *European Journal of Operational Research*, 117(2):340–354, 1999.
[žiūrėta 2021-01-16]. Prieiga per internetą:
<<https://www.sciencedirect.com/science/article/abs/pii/S0377221798002756>>.
- [Tou03] Stephen E. Toulmin. *The uses of argument*. Cambridge university press, 2003.
- [WD13] Kaitlynn M. Whitney & Charles B. Daniels. The root cause of failure in complex it projects: complexity itself. *Procedia Computer Science*, 20:325–330, 2013.
[žiūrėta 2021-05-14]. Prieiga per internetą:
<<https://www.sciencedirect.com/science/article/pii/S1877050913010806>>.
- [WL13] X. Wang & J. Li. Detecting communities by the core-vertex and intimate degree in complex networks. *Physica A: Statistical Mechanics and its Applications*, 392(10):2555–2563, 2013. [žiūrėta 2020-03-08]. Prieiga per internetą:
<<https://www.sciencedirect.com/science/article/pii/S0378437113000915>>.
- [XL05] W. Xia & G. Lee. Complexity of information systems development projects: conceptualization and measurement development. *Journal of management information systems*, 22(1):45–83, 2005. [žiūrėta 2020-02-10]. Prieiga per internetą:
<<https://www.tandfonline.com/doi/pdf/10.1080/07421222.2003.11045831>>.

Priedas nr. 1

Dekomponavimo kortų metodą realizuojančio įrankio metaduomenys

ID	Atributo pavadinimas	Pildymo eiliškumas	Kortos dalis	Bazinis atributas	Išvestinis atributas	Automatinis Atributas	Priklausomybės
M-1	Metaduomenys	1	Meta	+			–
K-1	Savybių sąrašas	2	Kairė	+			–
K-2	Identifikatorius	3	Kairė			+	–
K-3	Prioriteto reitingas	4	Kairė	+			K-1
K-4	Santykinis prioriteto reitingas	5	Kairė		+		K-3
K-5	Sąveikų matrica	6	Kairė	+			K-1
V-1	Sudedamųjų dalių sąrašas	7	Viršus	+			–
V-2	Identifikatorius	8	Viršus			+	–
V-3	Ryšių matrica	9	Viršus	+			V-1
C-1	Atsakomybių matrica	10	Centras	+			K-1, V-1
D-1	Atsakingų dalių kiekis	11	Dešinė		+		C-1
D-2	Atsakingų dalių įverčių suma	12	Dešinė		+		C-1
D-3	Atsakingų dalių įverčių vidurkis	13	Dešinė		+		D-1, D-2
D-4	Alternatyvų savybių atributai	32	Dešinė			+	–
A-1	Svarbos reitingas	14	Apačia		+		K-3, C-1
A-2	Santykinė svarba	15	Apačia		+		A-1
A-3	Atsakomybių kiekis	16	Apačia		+		C-1
A-4	Atsakomybių verčių suma	17	Apačia		+		C-1
A-5	Atsakomybių verčių vidurkis	18	Apačia		+		A-3, A-4
A-6	Ryšių kiekis	19	Apačia		+		V-3
A-7	Ryšių verčių suma	20	Apačia		+		V-3
A-8	Ryšių verčių vidurkis	21	Apačia		+		A6, A7
A-9	Sudedamųjų dalių kiekis	22	Apačia		+		V-1
A-10	Svarbos reitingų suma	23	Apačia		+		A-1
A-11	Svarbos reitingų mediana	24	Apačia		+		A-1
A-12	Santykinis svarbos rangas	25	Apačia		+		A-2
A-13	Atsakomybių kiekis	26	Apačia		+		A-3
A-14	Atsakomybių verčių suma	27	Apačia		+		A-4
A-15	Atsakomybių verčių vidurkio rangas	28	Apačia		+		A-5
A-16	Ryšių kiekis	29	Apačia		+		A-6
A-17	Ryšių verčių suma	30	Apačia		+		A-7
A-18	Ryšių kiekio rangas	31	Apačia		+		A-6
A-19	Alternatyvų elemento atributai	33	Apačia			+	–

44 pav. DK metodą realizuojančio įrankio metaduomenys