

Komponentinė informacinės sistemos dekompozicija

Lina BAGUŠYTĖ, Audronė LUPEIKIENĖ (MII)

el. paštas: audronel@ktl.mii.lt

Reziumė. Straipsnyje nagrinėjama atotrūkio tarp verslo ir kuriamos informacinės sistemos reikalavimų mažinimo problema. Parodoma, kad problemos sprendimas susijęs su naudojama analizės metodika. Aptariami analizės metu taikomi dekompozicijos metodai ir jų ypatumai. Parodoma, kad pasirenkant informacinės sistemos kūrimo metodiką, padaromi patys bendriausieji projektiniai sprendimai, ir informacinės sistemos reikalavimų formulavimas vykdomas kuriamos sistemos architektūros kontekste. Pateikiama informacinės sistemos funkcinį reikalavimų formulavimo, atsižvelgiant į verslo sistemos modelį, strategija. Jos taikymas leidžia dekomponuoti informacinę sistemą į sąveikaujančius komponentus.

Raktiniai žodžiai: atotrūkio tarp verslo ir informacinės sistemos mažinimas, informacinės sistemos dekompozicija, funkciniai reikalavimai.

Įvadas

Šiuolaikinė įmonė yra trijų lygmenų sistema, į vieningą visumą jungianti verslo, informacijos apdorojimo, programinės bei techninės įrangos komponentus. Informacinių sistemų (IS) ir jų programinės įrangos kūrimo metodai turi sudaryti prielaidas integruoti trijų lygmenų sistemas ir šitaip eliminuoti potencialų programinės įrangos neatitikimą realiems kompiuterizuojamo verslo poreikiams.

Teorija, kaip kurti integruotas, apjungiančias heterogeninius komponentus informacinės sistemas, yra nepakankamai išplėtotą. Akcentuojama [13], kad mažinant atotrūkį tarp verslo poreikių ir IS, reikalingi metodai, kurie leistų kiek įmanoma anksčiau atvaizduoti kompiuterizuojamus procesus į konkrečius IS komponentus ir nustatyti jų sąveiką. Šiai problemai darbe [13] siūloma naudoti metodiką, apimančią specifikuojamą kalbą GRL (angl. *Goal-oriented Requirement Language*) ir scenarijais grindžiamą architektūros notaciją UCM (angl. *Use Case Maps*). GRL kalba naudojama formuluoti tiek funkciniam, tiek nefunkciniam reikalavimams, bei pastariesiems operacionalizuoti. Šiame procese gauti IS projektiniai sprendimai yra aprašomi UCM scenarijais, atsakant ne tik „ką“ sistema daro, bet ir „kaip“. Tačiau ši metodika aprašoma reikiamų sukurti artefaktų terminais, nėra netgi euristinių taisyklių, kaip juos gauti vieną iš kito. Paslaugomis grindžiamos IS kūrimo paradigmos tikslas taip pat yra mažinti atotrūkį tarp verslo ir IT, nusakant su įmonės poreikiais suderintų informacijos apdorojimo paslaugų aibę [17]. Paslaugos inkapsuliuoja pakartotinai panaudojamas verslo funkcijas [9]. Tačiau informacijos apdorojimo paslaugų rinkinio, atitinkančio įmonės poreikius, identifikavimas nėra triviali užduotis. Servisinės analizės ir projektavimo metodikoje nenurodoma, kaip ir kokiais kriterijais vadovaujantis atliekamas verslo procesų atvaizdavimas į paslaugas, o paslaugų – į operacijas, kaip paslaugos turi būti apjungiamos į sudėtingas, siekiant užtikrinti reikiamą IS funkcionalumą.

Šiame darbe nagrinėjama funkcinių reikalavimų „nuleidimo“ iš verslo į informacijos apdorojimo lygmenį problema. Kitaip tariant, nagrinėjami būdai, kaip pereiti nuo verslo prie informacijos apdorojimo lygmens reikalavimų, išvengiant atotrūkio tarp verslo tikslų ir jiems palaikyti skirtų informacinių sistemų. Parodoma, kad problemos sprendimas susijęs su naudojama analizės metodika. Pateikiama informacinės sistemos funkcinių reikalavimų formulavimo, atsižvelgiant į verslo sistemos modelį, strategija. Jos taikymas leidžia dekomponuoti informacinę sistemą į sąveikaujančius komponentus.

Informacinės sistemos kūrimo ypatumai

Informacinių sistemų ir jų programinės įrangos kūrimo metodai turi sudaryti prielaidas integruoti trijų lygmenų komponentus ir šitaip eliminuoti potencialų programinės įrangos neatitikimą realiems kompiuterizuojamo verslo poreikiams. Tai reiškia, kad pirmajame IS kūrimo žingsnyje kompiuterizuojamos verslo sistemos tikslus reikia transformuoti į šiems tikslams pasiekti reikalingas informacines, skaičiuojamąsias ir komunikavimo paslaugas teikiančios informacinės sistemos reikalavimus. IS reikalavimai formuluojami prisilaikant pasirinktos metodikos. Parodysime, kad pasirenkant metodiką, padaromi patys bendriausieji projektiniai sprendimai, ir IS reikalavimų formulavimas vykdomas kuriamos sistemos architektūros kontekste.

1 lentelėje yra pateiktos informacinės sistemos kūrimo metodikų klasių charakteristikos. Verslo sistemos požiūriu informacinė sistema gali vykdyti tam tikras funkcijas, savarankiškai atsakyti už tam tikrų verslo problemų sprendimą ir tarpininkauti, kad būtų suteiktos reikiamos paslaugos verslo sistemai. Taigi, informacinės sistemos, priklausomai nuo verslo reikalavimų, turi turėti skirtingą pobūdį.

Jei pirmos kartos IS vykdė tam tikras verslui reikalingas funkcijas (automatizavo informacijos apdorojimą), tai antrosios kartos IS yra integruota sudėtinė verslo sistemos dalis ir yra kuriamos kaip aktyvios, savarankiškos, išskirstytomis paslaugomis besinaudojančios sistemos. Architektūros stilius apibūdina projektuojamų IS savybes ir ypatumus. Todėl, siekiant sukurti verslo tikslus padedančias pasiekti informacines

1 lentelė. Architektūros stiliaus sąlygojamas IS pobūdis

IS pobūdis verslo sistemos požiūriu	Funkcinis	Funkcinis	Funkcinis	Užduočių (angl. <i>task</i>)	Paslaugų
Metodika	Struktūrinė analizė ir projektavimas	Objektinė analizė ir projektavimas	Komponentinis kūrimas	Agentinė metodika	Servisinė analizė ir modeliavimas
Dekompozicija	Funkcinė	Objektinė	Komponentinė	Į užduotis	Į paslaugas
Architektūros stilius	Duomenų srautų stiliaus	Objektinio, sluoksninio stiliaus	Komponentinio stiliaus	Įvykiais valdomo, saugyklos stiliaus	Servisinio stiliaus

sistemas, tas pats architektūros stilius nėra tinkamas visoms IS klasėms realizuoti [3]. Iš to, kas pateikta, akivaizdu, kad norint sukurti skirtingus ypatumus turinčias IS, reikia naudoti skirtingas analizės ir projektavimo metodikas. Be to, antrosios kartos informacinėms sistemoms kurti struktūrinės analizės ir projektavimo metodikos yra nepakankamos, bet gali būti taikomos atskiriems IS komponentams arba žemesniesiems IS architektūros sluoksniams realizuoti.

Komponentinis kūrimas informacinę sistemą nagrinėja kaip nepriklausomų komponentų, sąveikaujančių per standartizuotus kontraktiškai specifikuotus interfeisus, bei realizuojamų nepriklausomai vienas nuo kito, visumą. IS komponentai turi tenkinti tam tikrus standartus, sudarančius galimybę juos jungti tarpusavyje, panaudojant leistinus jungimo būdus. Agentinės metodikos akcentuoja IS savarankiškumą, gebėjimą siekti nustatytų tikslų ir tinkamai funkcionuoti konkrečiose situacijose. Informacinė sistema nagrinėjama kaip sąveikaujančių: komunikujančių ir kartu vykdančių užduotis – sudėtinų dalių visuma. Servisinė analizė ir modeliavimas IS traktuoja kaip tarpininką, užtikrinantį visas verslui reikalingas paslaugas. Paslaugas gali įgyvendinti pati informacinė sistema arba teikti verslo sistemos išorėje funkcionuojanti sistema.

Bendruoju atveju, dekompozicija yra visumos skaidymas į paprastesnes sudedamąsias dalis. Apibrėžiant IS reikalavimus verslo sistemos modelio pagrindu, reikia atsakyti į klausimą „Kaip išskirti informacinės sistemos paprastesnes (vykdančias skaičiavimus, teikiančias informacines ir komunikavimo paslaugas) sudėtines dalis, kurios užtikrins realių verslo sistemos poreikių įgyvendinimą?“. Kitaip tariant, dekompozicija yra esminė IS reikalavimų sudarymo, o taip pat projektavimo proceso dalis [6, 16, 20].

Sistemoms analizuoti ir projektuoti naudojamos šių tipų dekompozicijos:

- funkcinė [22, 25],
- duomenų [12, 19],
- objektinė [4, 5, 8, 15, 18, 21]
- užduočių (angl. *task-oriented*) [23, 24],
- komponentinė [10, 11],
- paslaugų (angl. *service-oriented*) [1, 9, 14].

Trumpai apžvelgsime jų pagrindinius ypatumus. Atliekant funkcinę dekomponavimą identifikuojamos funkcijos, transformuojančios pradinį duomenį į rezultatą, sudaroma funkcijų hierarchija. Funkcinis projektavimas „paslepia“ rezultatų gavimo detales funkcijoje. Objektinio dekomponavimo atveju sistema skaidoma į objektus, ir informacinės sistemos funkcionalumas išreiškiamas tarpusavyje komunikujančių objektų, inkapsuliuojančių duomenis ir verslo logiką, rinkiniu. Objektai dažniausiai atspindi realaus pasaulio esybes. Kai svarbiausiu aspektu laikomi duomenys, pagrindinis dėmesys skiriamas verslo-informacinėms esybėms.

Užduotimis grindžiamoje dekompozicijoje vyrauja organizacinis požiūris į sistemą. Kitaip tariant, sistema suprantama kaip sąveikaujančių savarankiškų esybių, turinčių tam tikrus vaidmenis, visuma. Dekomponavimo metu nagrinėjami sistemos tikslai, kurių pagrindu išskiriamos su vaidmenimis siejamos užduotys.

Skiriami du dekomponavimo į komponentus metodai: nuolatinės rekursijos ir diskrečiosios rekursijos. Šie metodai nėra skirtingi – antrasis yra pirmojo specializacija. Taikant nuolatinę rekursiją, sistema skaidoma į komponentus iki norimo granuliaru-

mo ar sudėtingumo. Pats sistemos kūrėjas turi įvardinti, kiek lygmenų bus naudojama, koks lygmens vaidmuo sistemoje, kokios yra lygmeniui priklausančių komponentų savybės. Diskrečiosios rekursijos atveju įvertinama pakartotinio panaudojamumo galimybė. Sistema dekomponuojama į žinomus tam tikro granuliarumo komponentų tipus. Programinis komponentas, verslo komponentas, verslo komponentų sistema yra skirtingo granuliarumo komponentų pavyzdžiai.

Servisinė dekompozicija reiškia, kad akcentuojamas pakartotinis panaudojimas, sistemos plėtimas ir sudėtinių dalių tarpusavio sąveika. Paslaugos – tai būdai, kuriais naudotojo poreikiai susiejami su teikėjo galimybėmis. Skaidymas į paslaugas vykdomas remiantis įmonės veiklos modeliu, taikant „iš viršaus žemyn“ (vykdoma veiklos dekompozicija ir nustatomos verslo procesams reikalingos IS paslaugos), „iš apačios į viršų“ (nustatomos jau egzistuojančios galimybės) ir „iš vidaus į išorę“ (analizuojant tikslus, nustatomos dar trūkstamos paslaugos) technikas [9].

Siekiant sukurti verslo poreikius tenkinančią informacinę sistemą, jos reikalavimai formuluojami remiantis verslo sistemos modeliu. Tai sąlygoja, viena vertus, vieninga įmonės tris lygmenis apimanti kūrimo aplinka [7], kita vertus, integruotas IS pobūdis.

Informacinės sistemos funkciniai reikalavimai ir dekompozicija

Siekiant apibrėžti IS reikalavimus verslo sistemos modelio pagrindu, siūloma naudoti dvi technikas: Astrakan metodą [2] ir užduočių (angl. *use case*) diagramas. Astrakan metodas taikomas nustatyti, kokių paslaugų ir informacijos iš IS reikia verslo procesams, verslo objektams ir kitiems verslo elementams (jų nagrinėjamas rinkinys priklauso nuo pasirinktos metodikos). Kitaip sakant, metodas taikomas verslo sistemos sąveikai su informacine sistema nagrinėti. Užduotys aukščiausiam abstrakcijos lygmenyje apibrėžia IS funkcinis reikalavimus, ir kartu gali būti nagrinėjamos kaip pagrindas IS komponentinei dekompozicijai gauti.

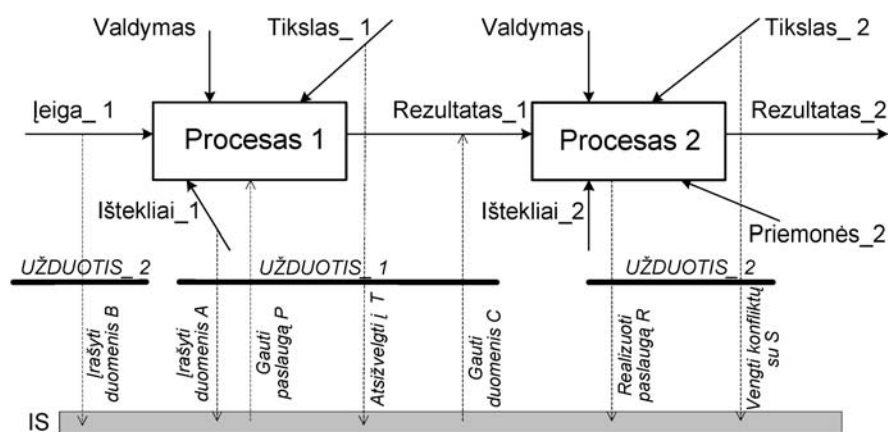
Siūloma tokia IS funkcinį reikalavimų formulavimo strategija (žr. 1 pav.):

1. Nagrinėjami verslo klientams (išoriniams ir vidiniams) paslaugas suteikti reikalingi pagrindiniai verslo procesai. Nustatoma, kokio palaikymo jiems reikia iš informacinės sistemos. Paveikslėlyje punktyrinėmis linijomis parodyta, kokia informacija (užklausa) ar paslaugos gaunamos/teikiamos informacinei sistemai.

2. Nagrinėjami su verslo procesais susiję (naudojami, kuriami, modifikuojami) verslo objektai. Kadangi IS funkcionuoja verslo sistemos aplinkoje, turi būti nustatoma, su kokiais verslo objektais ir kaip ji sąveikauja. Pavyzdžiui, reikia reaguoti į objekto pokyčius ar tam tikru laiku vykdyti reikiamus veiksmus. Paveikslėlyje verslo objektus sudaro verslo proceso įeiga, rezultatai, naudojami ištekliai, priemonės ar įrankiai. (Pastebėsime, kad jie gali būti aprašyti ir kitokio pavidalo diagrama.)

3. Nagrinėjami verslo sistemos tikslai, kuriuos realizuoja verslo procesai ir kuriuos įgyvendinti turi padėti kuriama IS. Ši analizė leidžia nustatyti IS reikalavimus, kurie nesusiję su klientų poreikių užtikrinimu. Pavyzdžiui, iš tikslų galima nustatyti vengtiną situaciją, galimas nesėkmes ir su tuo susijusį IS funkcionavimą. Kaip ir verslo objektų atveju, verslo sistemos tikslai gali būti aprašyti ir kitokio pavidalo diagrama.

4. Nagrinėjant 1–3 žingsniais apibrėžtą verslo ir informacinės sistemos sąveiką, užklausa ir paslaugos apjungiamos į logiškai susijusias grupes (1 pav. Užduotis_1 ir



1 pav. Informacinės sistemos užduočių identifikavimas.

Užduotis_2), kurios traktuojamos kaip informacinės sistemos užduotys. Nustatoma, ar šios užduotys tarpusavyje susijusios, jei taip, identifikuojami jų tarpusavio ryšiai.

Nustatytų IS užduočių ir ryšių tarp jų pagrindu apibrėžiami pagrindiniai IS komponentai. Gautieji aukščiausiojo abstrakcijos lygmens informacinės sistemos komponentai gali būti projektuojami pasirenkant kitas IS kūrimo metodikas ir atitinkamai kitus architektūros stilius. Lygiai taip pat, hierarchiškai, gali būti dekomponuojamos aukščiausiojo abstrakcijos lygmens komponentus siejančios jungtys.

Išvados

Siekiant eliminuoti potencialų informacinės sistemos neatitikimą realiems kompiuterizuojamo verslo poreikiams, pirmiausia reikia pasirinkti tinkamą IS kūrimo metodiką, nes IS reikalavimų formulavimas vykdomas architektūrinių sprendimų kontekste. Informacinių sistemų reikalavimai sudaromi, remiantis verslo sistemos modeliu. IS funkciniai reikalavimai nustatomi nagrinėjant verslo procesus, IS funkcionavimo aplinką ir verslo tikslus. Taikant pasiūlytą funkcinį reikalavimų nustatymų strategiją, informacinė sistema kartu dekomponuojama į sąveikaujančius komponentus.

Literatūra

1. A. Arsanjani, A. Allam, Service-oriented modelling and architecture for realization of SOA, in: *Proceedings of IEEE International Conference on Services Computing (SCC'06)*, vol. 521 (2006).
2. Astrakan, *The Astrakan Method*, Stockholm, Astrakan (1997).
3. L. Bagušytė, A. Lupeikienė, Verslo ir informacinių sistemų integravimas: architektūrinis aspektas, *Informacijos mokslai*, 155–162 (2007).
4. G. Booch, *Object-Oriented Design with Applications*, Benjamin Cummings (1991).
5. P. Coad, E. Yourdon, *Object-Oriented Analysis*, Yourdon Press Computing Series (1990).
6. A. Čaplinskas, *Programų sistemų inžinerijos pagrindai*, I dalis, Matematikos ir Informatikos institutas (1998).
7. A. Čaplinskas, A. Lupeikienė, O. Vasilecas, Shared conceptualisation of business systems, information systems and supporting software, in: H.-M. Haav, A. Kalja (Eds.), *Databases and Information Systems II*, Selected Papers, Kluwer Academic Publishers (2002), pp. 109–120.

8. O.J. Dahl, K.N. Simula, An Algol-based simulation language, *Communications of the ACM*, (9), 671–682 (1966).
9. M. Endrei *et al.*, *Patterns: Service-Oriented Architecture and Web Services*, International Technical Support Organization (2004).
10. D.F. D'Souza, A.C. Wills, *Objects, Components and Frameworks with UML: The Catalysis Approach*, Addison Wesley (1999).
11. P. Herzum, O. Sims, *Business Component Factory: a Comprehensive Overview of Business Component Development for the Enterprise*, John Wiley & Sons (2000).
12. M.A. Jackson, *Principles of Program Design*, Academic Press (1975).
13. L. Liu, E. Yu, From requirements to architectural design – using goals and scenarios, in: *Proceedings of the First International Workshop From Software Requirements to Architectures (STRAW'01)* (2001).
14. D. Manolescu, B. Lublinsky, *SOA Enterprise Patterns – Services, Orchestration and Beyond*, Morgan Kaufmann Publishers (2007).
15. B. Meyer, *Object-Oriented Software Construction*, Prentice Hall (1998).
16. D.L. Parnas, On the criteria to be used in decomposing systems into software modules, *Communications of ACM*, **15**(12), 1053–1058 (1972).
17. J.J. Ritsko, A. Birman, L. Cherbakov, G. Galambos, R. Harishankar, S. Kalyana, G. Rackham, Impact of service orientation at the business level, *IBM Systems Journal*, **44**(4), 653–668 (2005).
18. J. Rumbaugh, *Object-Oriented Modeling and Design*, Prentice Hall (1991).
19. D. Warnier, *Logical Construction of Programs*, Van Nostrand Reinhold Company (1974).
20. A.I. Wasserman, Toward a discipline of software engineering, *IEEE Software*, **13**(6), 23–31 (1996).
21. R. Wirfs-Brock, Object-oriented design: a responsibility-driven approach, in: *Proceedings of OOP-SLA'89* (1989), pp. 71–75.
22. N. Wirth, Program development by stepwise refinement, *Communications of the ACM*, **14**(4), 221–227 (1971).
23. M. Wooldridge, P. Ciancarini, Agent-oriented software engineering: the state of the art, in: *Agent-Oriented Software Engineering*, vol. 1957, Springer-Verlag Lecture Notes in AI (2001).
24. M. Wooldridge, N.R. Jennings, D. Kinny, The Gaia methodology for agent-oriented analysis and design, in: *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers (2000), pp. 285–312.
25. E. Yourdon, *Techniques of Program Structure and Design*, Prentice-Hall (1975).

SUMMARY

L. Bagušytė, A. Lupeikienė. Component-oriented information system decomposition

This paper analyses how to bridge the gap between business and supporting information system requirements. It is showed that solution of the problem depends on the analysis methodology of the enterprise system. Decomposition methods used during the analysis stage are discussed. The strategy how to formulate information system functional requirements, taking into account the business system model, is presented. Using the proposed strategy information system is decomposed to interacting components as well.

Keywords: bridging the gap between business and information system, information system decomposition, functional requirements.