

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA**

Vytautas Urbonas

**ORGANIZACIJOS ŽINIŲ VALDYMO
INFORMACINIŲ SISTEMŲ INTEGRACIJA**

Magistro darbas

**Vadovas
doc. dr. E. Karčiauskas**

KAUNAS, 2005

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA**

**TVIRTINU
Katedros vedėjas
dr. E. Bareiša
2005-05-23**

**ORGANIZACIJOS ŽINIŲ VALDYMO
INFORMACINIŲ SISTEMŲ INTEGRACIJA**

Programų inžinerijos magistro baigiamasis darbas

**Kalbos konsultantė
Lietuvių k. katedros lekt.
dr. J. Mikelionienė
2005-05-20**

**Recenzentas
doc. dr. R. Butleris
2005-05-20**

**Vadovas
doc. dr. E. Karčiauskas
2005-05-23**

**Atliko
IFM 9/2 gr. stud.
V. Urbonas
2005-05-18**

KAUNAS, 2005

KVALIFIKACINĖ KOMISIJA

Pirmininkas: Raimundas Stulpinas, UAB „Strauja“ generalinis direktorius, t.m.dr., doc.

Sekretorius: Kęstutis Motiejūnas, t.m.dr., doc.

Nariai: Eduardas Bareiša, t.m.dr., doc

Rimantas Butleris, t.m.dr., doc

Egidijus Kazanavičius, habil.m.dr., prof.

Dalius Rubliauskas, t.m.dr., doc

Rimantas Šeinauskas, habil.m.dr., prof.

Vytautas Štuikys, habil.m.dr., prof.

SUMMARY

This work is aimed to analyze enterprise knowledge management system integration assumptions, problems and their solutions at different complexity and quality levels.

In the beginning, brief description of application classification is provided. Assuming large number of specialized applications usage enterprise-wide, we come up with the need to integrate data, contained by those applications, as well as applications themselves. Data level, message level and process level integration principles and their application analysis is made.

Enterprise application integration refers to the plans, methods, and tools aimed at modernizing, consolidating, and coordinating the computer applications in an enterprise. Portals presented as tools for federating information in different sources and providing commonly used services (search engine or directory, news, email, forums and options for customization). Portal frameworks from different vendors was observed. SOA (Service Oriented Architecture) and BPEL (Business Process Execution Language) integration capabilities were analyzed.

Human Capital Resource Management information system is intended to store and process data, related to human capital across organization. Three layer logical implementation and web service interface usage makes integration possible at data, message and process levels.

TURINYS

1. Įvadas	7
1.1. Darbo tikslas.....	7
1.2. Pateiktis	7
2. Organizacijos žinių valdymo informacinių sistemų integracijos analizė.....	9
2.1. Organizacijos žinių valdymo informacinių sistemų dekompozicija	9
2.2. Organizacijos informacinių sistemų integracija.....	13
2.2.1. Aplikacijų integravimo problemos.....	13
2.2.2. Aplikacijų integravimo principai	15
2.3. Aplikacijų integravimo technologinės ir programinės priemonės	21
2.3.1. Portalai	21
2.3.1.1. Organizacijos žinių portalų techninė architektūra	22
2.3.1.2. IBM Organizacijos Informacinis Portalas (OIP)	24
2.3.1.3. Oracle portalų karkasas.....	27
2.3.1.4. Microsoft Share Point	28
2.3.2. Procesų lygmens integracijos technologijos	30
2.3.2.1. Į paslaugas orientuota architektūra	30
2.3.2.2. BPEL.....	33
3. Žmoginio kapitalo valdymo informacinės sistemos projektiniai sprendimai	35
3.1. Sistemos apibūdinimas.....	35
3.2. Sistemos architektūra	36
4. Žmoginio kapitalo valdymo informacinės sistemos projekcinio sprendimo tyrimas	43
4.1. Technologiniai sprendimai.....	43
4.2. Integracija.....	46
5. Išvados.....	47
LITERATŪRA	49
TERMINŲ IR SANTRUMPŲ ŽODYNAS	51
1 Priedas. BPEL technologijų sąsaja	53

Lentelių sąrašas

4.1 lentelė Techninė aplinka.....	43
-----------------------------------	----

Paveikslėlių sąrašas

2.1 pav. Sprendimo hierarchija	10
2.2 pav. Procesų lygmens integracijos schema	16
2.3 pav. Procesų lygmens aplikacijų integracijos, grįstos pranešimų tarpininko principu, architektūros schema.....	18
2.4 pav. Procesų lygmens aplikacijų integracijos, grįstos aplikacijų serverio principu, architektūros schema.....	20
2.5 pav. Korporacijos žinių portalo informacinių technologijų techninė architektūra [1].....	22
2.6 pav. IBM WebSphere II portalo architektūra.....	24
2.7 pav. Oracle portalo struktūra.....	27
2.8 pav. SharePoint serverio ir svetainių architektūra.....	29
2.9 pav. Žiniatinklio paslaugų (<i>web services</i>) dalyviai	31
2.10 pav. Paslaugų struktūra ir sąveika	32
2.11 pav. Paslaugos panaudojimo veiksmų seka.....	32
3.1 pav. Organizacijos rinkos vertės struktūra	35
3.2 pav. HCMIS konteksto diagrama	36
3.3 pav. HCMIS paketų diagrama.....	37
3.4 pav. WebSite paketo klasių diagrama	38
3.5 pav. BusinessFacade paketo klasių diagrama	39
3.6 pav. BussinessRules paketo klasių diagrama	39
3.7 pav. WebControls paketo klasių diagrama.....	40
3.8 pav. DataAccess paketo klasių diagrama	41
3.9 pav. HCMIS DB schema.....	42
4.1 pav. ADO.NET duomenų prieigos architektūra [8]	43
4.2 pav. Duomenų lentelės vaizdavimo ir tvarkymo komponentas (EGrid).....	44
4.3 pav. Kliento savybių įvedimas	45
4.4 pav. Kliento savybių šablonas.....	45
4.5 pav. Kliento savybių įvedimas pritaikius šabloną.....	46

1. ĮVADAS

Organizacijų veiklos kompiuterizavimo epochos pradžioje, atskiriems uždaviniams spręsti buvo kuriamos atskiros programinės priemonės. Reta organizacija veiklą kompiuterizavo kompleksiskai. Kompleksiniam veiklos kompiuterizavimui reikia ne tik didelių investicijų programinei įrangai sukurti, bet ir aiškių organizacijos veiklos procesų apibrėžimų, o šito dažnai organizacijos gyvavimo pradžioje labai trūksta.

Organizacijos veiklos procesų evoliucinė branda lemia adekvačią programinių priemonių prigimtį. Taip laikui bėgant sukuriamas visas programinių priemonių rinkinys atskiroms veiklos problemoms spręsti.

Šalia taktinių, į atskirų problemų sprendimą orientuotų programų inžinerijos sprendimų, šiandien aktualus tampa strateginis programų inžinerijos požiūris, leidžiantis efektyviau panaudoti programinius resursus. Atsirado naujas terminas – *korporacijos aplikacijų integracija (Enterprise Application Integration)* – kuri apima planus, metodus ir įrankius, skirtus korporacijos viduje esančioms aplikacijoms modernizuoti, sujungti ir koordinuoti.

Šiame darbe nagrinėjamos organizacijos žinių valdymo sistemų integracijos problemos bei jų sprendimo metodai, pateikiami atliktų eksperimentų rezultatai.

1.1. Darbo tikslas

Šio darbo tikslas – išnagrinėti organizacijos žinių valdymo sistemų integracijos prielaidas, problemas, įvardinti įvairaus sudėtingumo ir kokybinio lygmens problemos sprendimo variantus, pateikti projekcinį sprendimą ir išanalizuoti projekcinėje dalyje sukurtos informacinės sistemos integracijos poreikius bei galimybes.

1.2. Pateiktis

Darbo pradžioje aptariama visos organizacijos informacinių sistemų bazė. Iškeliama prielaida, kad žinios turi būti kompiuterizuotos norint priimti efektyvius, žiniomis grįstus sprendimus. Taip pat iškelta prielaida, kad žinių kompiuterizacijos procesas turi būti savaiminis, t. y. programinės priemonės turi būti kuo artimesnės veiklos procesams. Kadangi veiklos procesų kompiuterizavimas vyksta palaipsniui, tai sudaro prielaidas atsirasti dideliame atskirų specializuotų programų kiekiui organizacijos mastu.

Specializuotos programos sudaro galimybes savaiminiam duomenų, informacijos, žinių ir sprendimų kompiuterizavimo procesui. Tačiau siekiant strateginių tikslų, būtina sukurti priemones, leidžiančias surasti reikiamą informaciją arba atlikti veiksmus bet kuriame organizacijos kompiuterizuotame procese.

Šiuolaikinis problemos sprendimas yra korporacijos disponuojamų specializuotų programinių priemonių apjungimas (integracija) į vieną korporacijos aplikaciją. Darbe nagrinėjami integracijos modeliai, problematika, metodiniai ir programiniai sprendimai.

Projektinėje dalyje aprašyti svarbiausi projektiniai ir techniniai sukurto produkto – žmoginio kapitalo valdymo informacinės sistemos – aspektai.

2. ORGANIZACIJOS ŽINIŲ VALDYMO INFORMACINIŲ SISTEMŲ INTEGRACIJOS ANALIZĖ

Žinios šiuolaikinėje organizacijoje gali atspindėti ne tik atskiras funkcines organizacijos veiklos sritis, bet ir visos organizacijos mastu sukauptą patirtį („*know-what*“, „*know-how*“, „*know-why*“). Šios žinios leidžia vadovams ir analitikams atrasti silpnąsias vietas veikloje. Paprastai darbus su žiniomis (*knowledge work*) atlieka didelę kompetenciją atitinkamose funkcinėse srityse (strateginis valdymas, rinkotyra, gamyba, apskaita ir finansai, žmoniškieji ištekliai) turintis personalas. Darbas su žiniomis gali būti apibrėžtas kaip žinių įgijimas, sukūrimas, paruošimas ir taikymas.

Reikia pažymėti, kad neegzistuoja vien programinių ir techninių priemonių, užtikrinančių visapusišką darbą su žiniomis. Norint sukurti žinių valdymo infrastruktūrą įmonės mastu, reikia suderinti skirtingas technologijas. Jei įmonė nori atskleisti kolektyvines žinias įmonės mastu, ji turi sukurti programines ir technines priemones, o taip pat suderinti įmonės veiklos procesus su šia žinių valdymo technologija. Jei darbuotojai nedirba bendradarbiaujančioje aplinkoje, arba jei jų veikloje nėra numatyta priemonių žinių apsaugimui, vien žinių valdymo sistemų technologijomis ši problema nebus išspręsta.

Žinių valdymo sistemos skirtos sudaryti sąlygas profesionaliems ir techniniams darbuotojams kaupti žinias, organizuoti jas taip, kad jos būtų lengvai pasiekiamos ir efektyviai pritaikytos įmonės veikloje. Darbo su žiniomis procesas gali būti grindžiamas *žinių paieška esamuose duomenų šaltiniuose*, *žinių sukūrimu* iš dar nenaudotų žinių šaltinių ir *žinių pakartotiniu panaudojimu*. Žinios taip pat gali būti sukurtos organizacijos išorėje (pavyzdžiui, žinias apie finansinę atskaitomybę gali perteikti auditoriai).

Šiuolaikinėse žinių valdymo informacinėse sistemose dažnai integruojami išbaigti programiniai produktai (COTS – *Common off the shelf*): grupinio darbo, dokumentų valdymo, elektroninio pašto, reliacinių duomenų bazių, darbų proceso organizavimo programiniai produktai, intraneto paieškos varikliai, žinių suradimo (*knowledge discovery*) ir duomenų analizės (*data mining*) programinė įranga. Taip pat sistemose gali būti panaudota OLAP, statistinės analizės ir GUI/4GL programiniai paketai, palengvinantys sprendimų priėmimui reikalingos informacijos pateikimą.

2.1. Organizacijos žinių valdymo informacinių sistemų dekompozicija

Šiuolaikinėje organizacijoje siekiant gerų rezultatų, būtina kaupti ir perduoti žinias. Žinių kaupimą ir perdavimą įgalina naujausios techninės ir programinės priemonės.

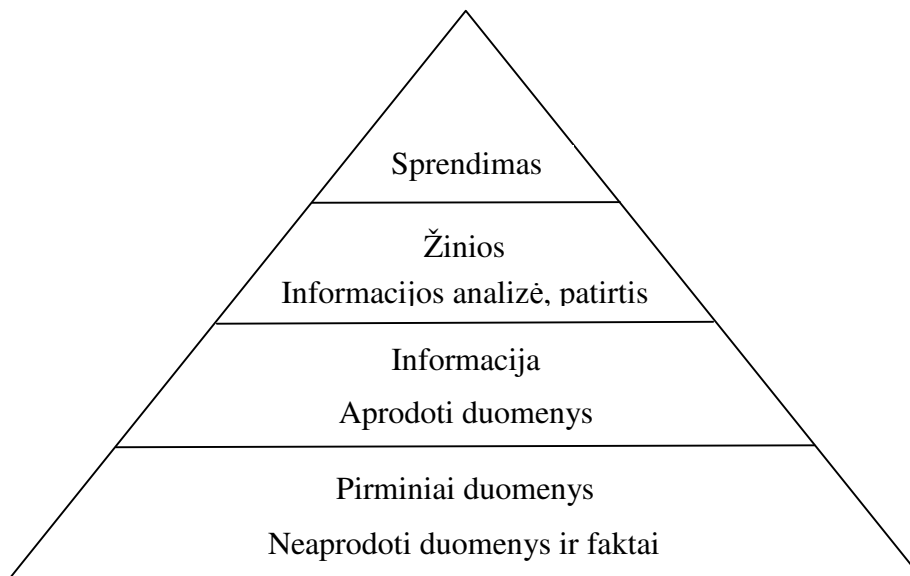
Svarbu pažymėti tai, kad norint žinių kaupimo procesą padaryti efektyviu, žinių kaupimas turi būti kuo artimesnis savaiminiam procesui. Žinių pagrindą turėtų sudaryti iš veiklos proceso išplaukiančios žinios, o ne atskirai „įvedamos“ žinios.

Kasdieniam darbe darbuotojas priima sprendimus. Sprendimo priėmimo ciklas yra toks:

1. Duomenų (informacijos) surinkimas;
2. Tyrimas ir bendradarbiavimas;
3. Veikimas (sprendimų priėmimas).

Kiekviename sprendimų priėmimo etape panaudojamos ir sukuriamos žinios turėtų patekti į sistemą, savo ruožtu sistema turėtų užtikrinti žinių panaudojimo galimybes.

Efektyvus sprendimas grindžiamas tam tikrais duomenimis, informacija ir žiniomis, reikalingomis problemos sprendimui. Tačiau kiekviena iš šių dedamųjų turi skirtingą rolę sprendimo priėmimo procese (2.1 pav.). Duomenys gaunami stebint ir dokumentuojant faktus; informacija gaunama analizuojant ir apdorojant duomenis; žinios atsiranda analizuojant informaciją, įgavus patirties ir supratimo.



2.1 pav. Sprendimo hierarchija

Idealiu atveju, visi sprendimo priėmimo metu naudoti duomenys ir rezultatai turėtų patekti į kompiuterines laikmenas, kad ateityje sukaupti duomenys, informacija, žinios ir priimti sprendimai galėtų būti panaudoti vėliau. Praktiškai turbūt neįmanoma pradedančiai veiklą įmonei taip tiksliai apibrėžti veiklos taisykles, kuriose aiškiai matytųsi, kurie duomenys ir kur bus panaudoti. Dažnesnis sprendimas – pasirenkamas procesas, kurį norima pagerinti ar automatizuoti, ir tada jau specifikuojami išoriniai ir vidiniai duomenų srautai.

Reikia pažymėti, kad tie patys duomenys, organizacijos informacinėms sistemoms evoliucionuojant, iš pradžių gali būti kaip pirminiai duomenys, kompiuterizavus papildomus veiklos procesus, gali tapti „išvestiniais“ duomenimis. (Scenarijus: įmonės projektų registre, užbaigus projektą, registruojamas projektas ir projekto apimtys. Projekto apimtys šiuo atveju kompiuterizuotų procesų prasme yra pirminis

duomuo, o įvertinimą atlieka projektų vadovai pasibaigus projektui. Vėliau, įmonei įdiegus darbo laiko apskaitos sistemą, darbuotojai patys registruoja kiek laiko kasdien sugaišo vykdydami projektą, o projekto apimtys jau gali būti apskaičiuotos, taigi yra informacinio lygmens).

Pagrindinis akcentas kompiuterizuojant organizacijos veiklos procesus, yra veiklos ir kompiuterinio proceso panašumas, t. y. kompiuterinėje sistemoje veiklos procesas turi būti kuo geriau atkartotas. Didelėse organizacijose yra didelė veiklų įvairovė, kuri lemia taip pat didelę kompiuterinių programų įvairovę.

Specifiniai produktai yra patogūs galutiniam naudotojui (įpročiai, kompaktiškos programos, specializuoti poreikiai), tačiau savo ruožtu sąlygoja standartizacijos ir vienodumo problemas. Korporacijų aplikacijų integravimo technologijos leidžia suderinti šias problemas. ***Korporacijos aplikacijų integracija (Enterprise Application Integration)*** apima planus, metodus ir įrankius, skirtus korporacijos viduje esančioms aplikacijoms modernizuoti, sujungti ir koordinuoti.

Žinių valdymo sistemas sudaro tokie įrankiai:

1. Specializuotos programos;
2. Infrastruktūrinės programinės priemonės;
3. Portalai;
4. Žinių struktūrizavimo priemonės (*Knowledge Extraction Tools*);
5. Intraneto paieškos varikliai;
6. OLAP ;
7. Statistinės analizės priemonės.

Specializuotos programos

Specializuotos programos – tai konkrečiam uždaviniui atlikti skirtos programos. Jos užtikrina savaiminį nuolatinį sistemos duomenų atnaujinimą, kadangi jomis darbuotojai naudojami natūraliai. Programose naudojami struktūrizuoti duomenys. Šie duomenys yra tiksliausi, todėl jų apibendrinimui ir analizei gali būti naudojamos tiek iš anksto numatytos analitinės ataskaitos, tiek apibendrinančios statistinės analizės priemonės.

Infrastruktūrinės programinės priemonės

Tai nuo taikomosios srities nepriklausančios programinės priemonės. Šios priemonės gali būti naudojamos tiek nepriklausomai, tiek kaip specializuotos programos sudėtinė dalis.

Portalai

Portalas – tai visų įmonės turimų sistemų „fasadas“. Jis projektuojamas taip, kad sujungtų skirtingus duomenų šaltinius ir sistemas, ir pateiktų vientisą, personalizuotą informaciją.

Žinių struktūrizavimo priemonės (*Knowledge Extraction Tools*)

Dažniausiai organizacija žinių valdymo priemonės pradeda taikyti ne iškart pradėjusios veiklą. Pirmas uždavinys, su kuriuo susiduriama pradėdant naudoti žinių valdymo informacines sistemas, yra pradinių duomenų užpildymas. Rankinis duomenų užpildymo metodas turbūt yra tiksliausias, tačiau praktikoje taikomas retai dėl didelių duomenų apimčių. Automatiniam duomenų užpildymui naudojamos žinių struktūrizavimo priemonės. Nestruktūrizuotuose tekstiniuose duomenų šaltinius atliekama lingvistinė analizė: skaidomas dokumentas (antraštės, paragrafai, sakiniai, žodžiai), nustatoma kalbinė kiekvieno žodžio rolė kiekviename sakinyje (daiktavardis, būdvardis, veiksmažodis). Ši daugiau struktūrizuota informacija gali būti indeksuota paieškos varikliuose, klasifikuojama ir pan.

Intraneto paieškos varikliai

Žinių valdymo programinė įranga, naudojanti intraneto paieškos variklius yra plačiausiai naudojama priemonė informacijos organizacijos tinkle esančios informacijos paieškai. Žinios išgaunamos analizuojant intranete esančius dokumentus, žodynus, elektroninius laiškus ir kitokio pobūdžio informaciją.

Žinių išgavimo priemonės (*Knowledge Discovery or Data Mining*)

Žinių išgavimo programinė įranga leidžia atrasti duomenų struktūrą, tendencijas ir taisykles esamu laiko momentu bei atlikti prognozes.

Duomenų bazių valdymo sistemų gamintojai papildė savo produktus duomenų išgavimo (*data mining*) galimybėmis, tuo būdu duomenų išgavimo metodams suteikdami sąryšinių duomenų bazių duomenų manipuliacijos našumą.

OLAP (*On-Line Analytical Processing*)

OLAP technologijos leidžia greitai ir įvairiais pjūviais peržiūrėti ir analizuoti didelius informacijos kiekius, atlikti prognozavimą, naudojant tam pritaikytą duomenų modelį, kuris atspindi realų daugiamatį organizacijos veiklos vaizdą. Daugiamačiam duomenų modeliui atvaizduoti, OLAP sistemose naudojami daugiamatiai duomenų kubai. Daugiamatės sistemos koordinacinių ašimis yra pagrindiniai analizuojamo proceso atributai.

Statistinės analizės priemonės

Nors šiuolaikiniai programinės įrangos paketai realizuoja kai kurias statistines funkcijas, išsamiai statistinei analizei atlikti naudojamos specializuotos statistinės analizės sistemos. Dažna veiklos situacija, kai suformavus duomenų imtį, yra modeliuojami veiklos rezultatai esant pasikeitusioms sąlygoms. Tokiu

būdu gali būti patikrinami hipotetiniai sprendimai. Įrankių dėka, šiems veiksmams atlikti nereikia gilių programavimo žinių, todėl juos gali atlikti veiklos specialistai be programuotojų pagalbos.

2.2. Organizacijos informacinių sistemų integracija

Ankstesniuose skyriuose apibrėžėme žinių šaltinių ir žinių tvarkymo sistemų įvairovę. Šiame skyriuje nagrinėsime korporacijos aplikacijų integravimo problemas, integravimo technologijas.

Integruojamas sistemas galime interpretuoti kaip galutinės sistemos sudedamąsias dalis – komponentus. Todėl sistemų integracijos kontekste sąvokos „komponentas“, „aplikacija“, „programinis produktas“ vartojamos kaip sinonimai.

2.2.1. Aplikacijų integravimo problemos

Norint efektyviau išnaudoti programinius produktus, juos reikia susieti. Paprastai atskirose problemose dalyvaujantys duomenys, veikloje turi loginę sąsają, todėl analogiška sąsaja įmanoma ir problemų sprendimui taikomose kompiuterinėse programose.

Atskiroms integravimo problemoms egzistuoja skirtingi sprendimai. CeBASE¹ organizacija nurodo penkias bendras integravimo technologijas – strategijas :

¹ Center for Empirically-Based Software Engineering (CeBASE). CeBASE analizuoja empirinius modelius ir pateikia empiriškai patikrintas gaires technologijų ir modelių parinkimui, rekomendacijas programų inžinerijos tyrimams ir mokymams.

1. Parametrizavimas. Programinių produktų kūrėjai numato galimybę keisti komponento savybes parametrų pagalba nekeičiant paties komponento. Ši strategija dar vadinama „pilkos dėžės pakartotinis panaudojimas“ (*grey-box reuse*).
2. Komponento keitimas. Jei programinio produkto išeities tekstas prieinamas, programuotojai gali modifikuoti komponentą ir sukurti integravimui reikalingas sąsajas. Ši strategija dar vadinama „baltos dėžės pakartotinis panaudojimas“ (*white-box reuse*).
3. Perkūrimas. Jei programinis produktas netenkina tam tikrų sąlygų, norint apeiti problemas, jis yra perkuriamas.
4. Tarpinė programinė įranga (*glueware*). Kai kurios problemos, tokios kaip skirtingi parametrų tipai ar pavadinimai, gali būti išspręstos panaudojant sąlyginai paprastus kodo fragmentus, kurie patalpinami tarp sąveikaujančių komponentų. Atskiras tarpinės programinės įrangos atvejis – komponento aplankas (*wrapper*), kuris apgaubia komponentą, ir koordinuoja komponento sąveiką su kitais komponentais.
5. Architektūriniai pakeitimai. Kai kurių integravimo problemų sprendimas reikalauja visos sistemos architektūros pakeitimo. Pavyzdžiui, CORBA komponentų integravimas gali sąlygoti SOM (*System Object Model* – architektūrinis sprendimas, leidžiantis skirtingoms aplikacijoms naudoti kompiliuotus komponentus) architektūrinio šablono pritaikymą visos sistemos mastu.

Savo ruožtu, komponentų integravimo metu kylančios problemos gali būti skirstomos:

1. Funkcines problemas. Šio tipo problemos susijusios su neteisingu funkcionalumu, ir integracijos strategija dažniausiai bus komponento perkūrimas arba keitimas ištaisant klaidingai veikiančią komponentą.
2. Nefuncines problemas. Nefunkcinės problemos: pernešamumas, palaikomumas, panaudojamumas, greitaeigiškumas ir pan. Integracijos strategija – komponento perkūrimas.
3. Architektūrinio stiliaus problemos. Tai integravimo problemos, kylančios norint apjungti komponentus su skirtingais programavimo, valdymo stiliais, kurie gali įtakoti komponentų sąveiką. Šio tipo problemos gali būti sprendžiamos panaudojant tarpinę programinę įrangą – apvalkalus arba adapterius, kurie leidžia suvienodinti architektūrinius stilius.
4. Architektūrinės problemos. Šio tipo problemos – tai konfliktai tarp sistemos komponentų (*deadlock*), ne adekvačios duomenų struktūros ir pan. Integravimo strategija šiuo atveju gali būti

sistemos konfigūracijos pakeitimai arba kai kuriais atvejais – tarpinės programinės įrangos panaudojimas.

5. Sąsajos problemos. Šio tipo problemos kyla kai bandoma integruoti komponentus su skirtingom sąsajom, arba komponentų sąveikos rezultate išryškėja defektai. Galima integravimo strategija – tarpinės programinės įrangos panaudojimas.

2.2.2. Aplikacijų integravimo principai

Aplikacijų integracija neturėtų būti taikoma „mados principu“. Integracijos poreikį turėtų lemti veiklos poreikiai ir galimybės.

Paprastai, išskiriami šie aplikacijų integracijos lygmenys:

Duomenų lygmuo. Tai yra duomenų integracija, kuri reiškia, kad aplikacijos naudojama duomenų saugykla, leidžiama naudotis (labai dažnai – tik peržiūros režimu) ir kitoms aplikacijoms. Tokios duomenų integracijos plitimą ypač paspartino duomenų bazių valdymo sistemų standartizacija ir protokolai (ODBC, JDBC), duomenų replikavimo priemonės ir duomenų saugyklos (*Data Warehouse*). Vienas labiausiai paplitusių šio tipo integracijos pavyzdžių – statistinės analizės ir daugiasluoksnės analizės programinės priemonės (pvz. Business Objects), kurios pagrįstos technologinėmis galimybėmis prisijungti prie skirtingų duomenų šaltinių.

Pranešimų lygmuo. Integracija grįsta pasikeitimu pranešimais tarp skirtingų aplikacijų. Pranešimų siuntimas ir priėmimas įtakoja pačias aplikacijas, kadangi jose turi būti numatytos (t. y. dažniausiai atskirai suprogramuotos) sąsajos. Šio tipo integracijos plitimą paspartino OLE technologija, ir iš jos evoliucionavusios COM, DCOM technologijos. Pavyzdys – Microsoft Word dokumento generavimas panaudojant OLE sąsają (tiesa, šiuo atveju realizuota vienakryptė integracija).

Procesų lygmuo. Ši integracija grindžiama veiklos proceso aprašymu, ir esamų aplikacijų prijungimą prie aprašyto proceso. Galima teigti, kad procesų lygmens integracija yra tam tikras pranešimų integracijos lygmens išplėtimas, kadangi realus duomenų apsikeitimas vyksta siunčiant pranešimus, tačiau tarpinė programinė įranga (*middleware*) veikia kaip darbų eigos (*workflow*) variklis, t. y. organizuoja visą pranešimų apsikeitimo procesą.

Pasirenkant integracijos lygmenį reikia atsižvelgti į specifinius organizacijos poreikius ir pasverti realizacijos kainą. Nors procesų lygmens integracija yra pažangus techninis sprendimas, kiti veiklos faktoriai taip pat turi būti įvertinti.

Jei organizacijos tikslas – aplikacijų vidutinio sudėtingumo integracijos uždaviniai ir ribotas biudžetas, turbūt geriausias variantas yra aplikacijų integracija duomenų lygmenyje. Paprastai šis

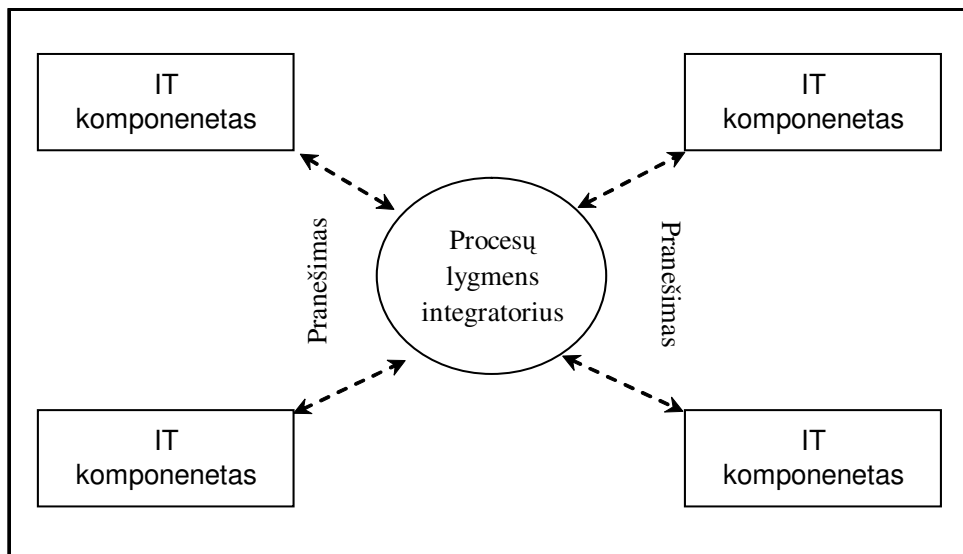
sprendimas būna pigiausias, nors išplėtimas ir palaikymas gali būti sudėtingas ir brangus. Šiuo būdu gali būti sprendžiami įmonės vidinės integracijos uždaviniai, tačiau integracija su išorinėmis organizacijomis yra pakankamai sudėtinga (B2B – *Bussines to Bussines*).

Pranešimų lygmens integracija taikoma nedidelėms organizacijoms, kuriose veiklos procesų kiekis nėra didelis, arba veiklos procesai nėra gerai apibrėžti.

Pranešimų lygmens integracija gali būti arba kaip laikinas sprendimas prieš pereinant į procesų lygmens integraciją, kadangi iš tokio tipo integracijos perėjimas į procesų lygmens integraciją yra pakankamai paprastas.

Procesų lygmens korporacijų aplikacijų integracija

Korporacijos aplikacijų integracija procesų lygmenyje vaizduojama kaip susijusių veiklos procesų aibė. 2.2 pav. matom, kad procesų lygmens korporacijų aplikacijų integracija yra centrinis taškas, palaikantis veiklos procesų vykdymą. IT komponentai apdoroja realius duomenis ir skaičiuoja, o informacijos apsikeitimas atliekamas siunčiant pranešimus. Procesų lygmens aplikacijų integracija reguliuoja pranešimų eiliškumą, sąlyginių ir loginių taisyklių sukūrimą ir tuo būdu apibrėžia bendrą procesų vykdymą.



2.2 pav. Procesų lygmens integracijos schema

Procesų lygmens aplikacijų integracijos projektavimas

Įgyvendinant aplikacijų integraciją procesų lygmenyje, būtina suprasti veiklos procesus, taisykles, įvykius, ir veiklos procese dalyvaujančius programinius komponentus bei duomenis, kuriais turi keistis programiniai komponentai.

Procesų lygmens aplikacijų projektavimo metu reikia apibrėžti:

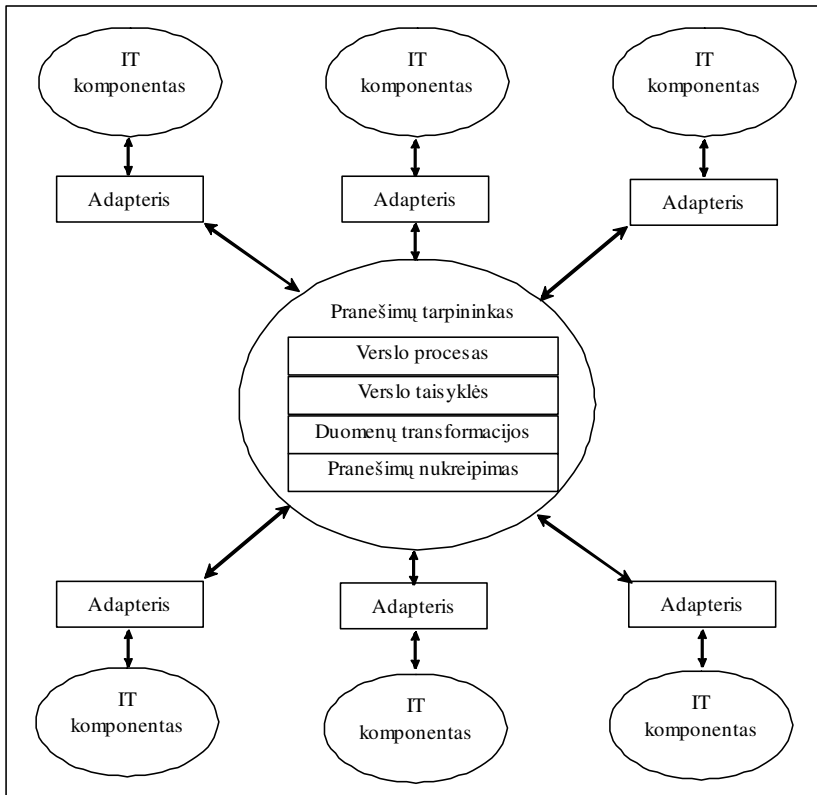
1. **Veiklos procesus.** Apibrėžiamos projekto apimtys, įvardijant kokie veiklos procesai turi būti sukurti arba palaikomi integruotoje korporacijoje.
2. **Veiklos įvykius.** Apibrėžiami veiklos įvykiai, kurie iššaukia veiklos procesus.
3. **Proceso dalyvius.** Įvardijami programiniai komponentai, dalyvaujantys veiklos procese.
4. **Proceso taisyklės ir pranešimai.** Veiklos taisyklės (sąlyginės išraiškos) apibrėžia programinių komponentų iškvietimo eiliškumą. Kartu su taisyklėmis, turi būti apibrėžti ir pranešimai, užtikrinantys duomenų apsikeitimą tarp programinių komponentų ir korporacijos aplikacijų integravimo variklio.
5. **Pranešimų turinys.** Po to, kai jau yra sukurtas pranešimų sąrašas, reikia apibrėžti kiekvieno pranešimo duomenų reikalavimus. Visi pranešimų apibrėžimai sudaro organizacijos pranešimų žodyną, kuris naudojamas organizacijos mastu.

Pranešimų turiniui fiksuoti ir vaizduoti rekomenduojama naudoti XML. XML yra defacto standartas sudėtingų duomenų aprašymui, kadangi nepriklauso nuo platformos ir yra lengvai skaitomas. Be to, XML teikia duomenų transformavimo galimybes, kurios leidžia standartizuoti pranešimų transformacijas tarp pranešimų siuntėjo ir gavėjo.

Proceso lygmens aplikacijų integracijos realizacija.

Labiausiai paplitę du realizacijos principai: pranešimų tarpininko principas ir aplikacijų serverio principas.

Pranešimų tarpininko (*Message Broker*) principas



2.3 pav. Procesų lygmens aplikacijų integracijos, grįstos pranešimų tarpininko principu, architektūros schema.

2.3 pav. matome, kad pranešimų tarpininkas yra visos sistemos ašis. Jo funkcijos:

Pranešimų nukreipimas. Visas pranešimų apsikeitimas vyksta tarp programinių komponentų ir pranešimų tarpininko. Kai reikia, pranešimų tarpininkas perduoda pranešimą kitiems komponentams.

Duomenų transformavimas. Pranešimų tarpininkas atlieka duomenų transformacijas, kad kiekvienam komponentui būtų pateikiami duomenys tokiu formatu, kurį komponentas „supranta“. Dažniausiai transformacijos sukuriamos panaudojant grafinius naudotojo sąsajos įrankius.

Veiklos taisyklių realizacija. Veiklos taisyklės reguliuoja veiklos proceso eigą. Pranešimų tarpininkai dažniausiai turi grafines naudotojo sąsajos priemones, leidžiančias vizualiai kurti veiklos taisykles. Sudėtingesnėms veiklos taisyklėms realizuoti galima panaudoti JAVA klases. JAVA klasėmis realizuotose taisyklėse galima panaudoti bet kokio tipo skaičiavimus, duomenų bazių prieigas ir pan., tačiau jos neturėtų pakeisti konkrečių komponentų funkcionalumo.

Veiklos proceso palaikymas. Pranešimų tarpininkas atsakingas už veiklos proceso palaikymą tam skirtų grafinių naudotojo sąsajos priemonių.

Programiniai komponentai dažniausiai prijungiami prie pranešimų tarpininko panaudojant adapterius. Šiame kontekste adapteris – tai sluoksnis tarp pranešimų tarpininko ir aplikacijų. Adapteris nustato atitikmenis tarp integruotos aplikacijos ir standartinės (*native*) pranešimų tarpininko sąsajos.

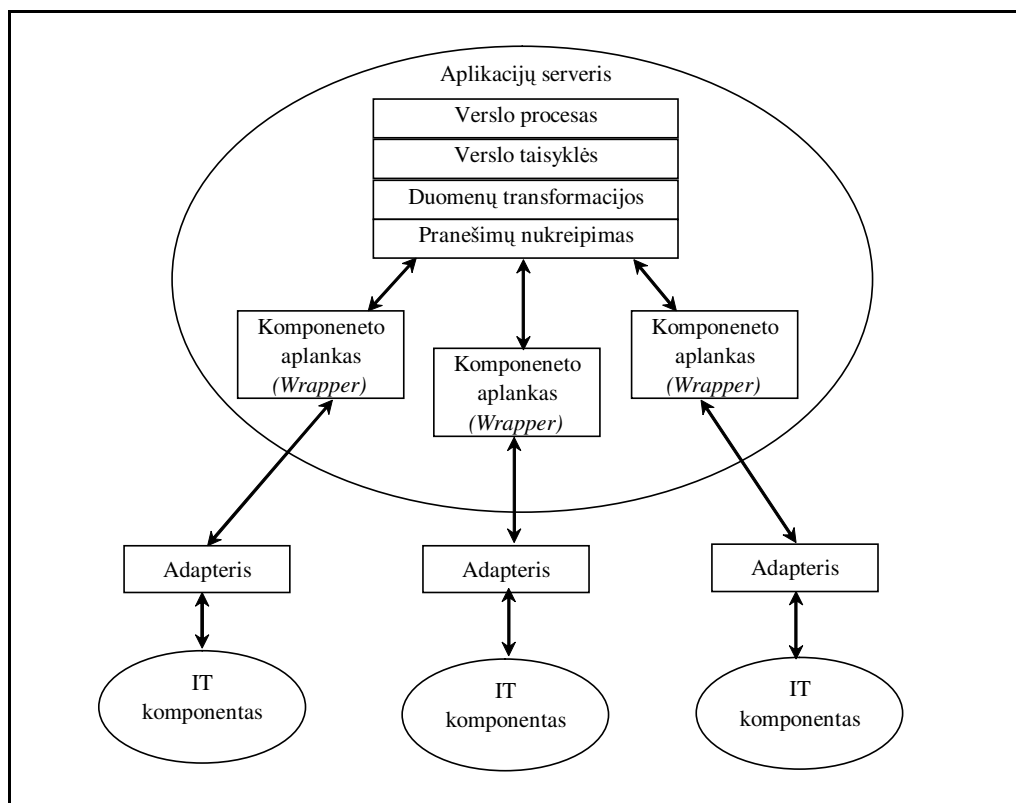
Šios architektūros privalumai:

1. Kiekvienas programinis komponentas sujungtas tik su pranešimų brokeriu, o ne tarpusavyje
2. Pranešimų apsikeitimas vyksta asinchroniniu būdu
3. Pranešimų tarpininkas atlieka visas duomenų transformacijas, todėl ši operacija atliekama centralizuotai
4. Pranešimų tarpininkas turi grafinę naudotojo sąsają veiklos taisyklių sukūrimui, todėl jų valdymas ir pakeitimai yra paprastesni
5. Garantuotas pranešimo perdavimo mechanizmas, realizuotas daugumoje pranešimų tarpininkų, yra svarbus momentas užtikrinant duomenų pilnumą ir neprieštarinumą
6. Standartiniai adapteriai ir jungtys palengvina sistemos konstravimą

Šios architektūros trūkumai:

1. Kadangi pranešimų apsikeitimas atliekamas asinchroniškai, viso proceso transakcijos nepalaikomos, todėl kartais reikia sukurti kompensuojančius procesus (kompensuojantis procesas – tai toks procesas, kurį įvykdžius po pirminio proceso, atstatoma sistemos būseną, buvusi prieš pirminio proceso vykdymą, pvz. Įterpimas ir ištrynimasis yra kompensuojančios funkcijos).
2. Saugumo realizacija sudėtinga.

Aplikacijų serverio principas



2.4 pav. Procesų lygmens aplikacijų integracijos, grįstos aplikacijų serverio principu, architektūros schema

2.4 pav. matome, kad visos sistemos ašimi yra aplikacijų serveris. Komponentai aplikacijų serveryje prijungiami panaudojant atitinkamus komponentų aplankus (*component wrapper*). Žiūrint iš aplikacijų serverio perspektyvos, procesas vyksta per aplikacijų serverio viduje esančius komponentus (t. y. jų aplankus), nors iš tikrųjų, komponentų aplankai tėra realių komponentų fasadas. Adapterio ir aplanko pora sudaro tam tikrą abstrakcijos lygmenį, kurio dėka komunikacija tarp programinių komponentų atliekama tarsi komponentai būtų vykdomi aplikacijų serverio aplinkoje.

Aplikacijų serverio funkcijos:

Pranešimų nukreipimas. Visi programiniai komponentai prijungti tik prie aplikacijų serverio, kuris atsakingas už reikiamo komponento iškvietimą reikiamu laiku.

Duomenų transformavimas. Aplikacijų serveris atlieka duomenų transformacijas, kad kiekvienam komponentui būtų pateikiami duomenys tokiu formatu, kurį komponentas „supranta“

Veiklos taisyklių realizacija. Aplikacijų serveryje realizuojamos visos veiklos taisyklės.

Šio tipo architektūra – tai žingsnis standartizacijos link. J2EE platforma apibrėžia standartinę jungiklio architektūrą (*connector architecture*) J2EE sujungimui su heterogeninėmis korporacijos informacinėmis sistemomis, tokiomis kaip ERP (*Enterprise Resource Planning*) sistemomis, transakcijų

valdymo monitoriais ir pan. Jungiklio architektūra numato visą rinkinį išplečiamų, saugių ir transakcijas palaikančių mechanizmų, kurie užtikrina korporacijų informacinių sistemų integraciją aplikacijų serveryje.

Jungiklio architektūra taip pat leidžia sistemos kūrėjui sukurti standartinį adapterį ir aplanką savo kuriamai sistemai. Aplikacijų serveryje įdiegtas aplankas per adapterį susijungia su sistema.

Šios architektūros privalumai:

1. Kiekvienas programinis komponentas sujungtas tik su aplikacijų serveriu, o ne tarpusavyje.
2. Aplikacijų serveris atlieka visas duomenų transformacijas, todėl ši operacija atliekama centralizuotai.
3. Visos reikalingos jungtys gali būti realizuotos sukuriant adapterius ir aplankus.
4. Viso proceso transakcijos gali būti realizuotos, jei komponentai palaiko transakcijas.
5. Bendras integruotų aplikacijų sistemos išplėčiamumas priklauso tik nuo atskirų komponentų (aplikacijų) išplėčiamumo.

Šios architektūros trūkumai:

1. Visas komunikavimas vyksta sinchroniškai, todėl ilgai trunkantis procesas gali išnaudoti aplikacijų serverio resursus.
2. Mažiau priemonių veiklos procesų, taisyklių ir duomenų transformacijų palaikymui.

Norint atlikti korporacijos aplikacijų integraciją procesų lygmenyje, reikia apibrėžti vidinius korporacijos veiklos procesus. Šis integracijos principas ne tik išsprendžia integravimo problemas, bet dar ir leidžia suvienodinti veiklos procesus visoje organizacijoje.

2.3. Aplikacijų integravimo technologinės ir programinės priemonės

Šiame skyriuje aptarsime dviejų tipų integravimo priemones: portalus ir veiklos procesų valdymo variklius.

2.3.1. Portalai

Kai kurie gerai žinomi programinių produktų ir paslaugų teikėjai, tokie kaip Oracle, IBM ir Microsoft šalia išbaigtų programinių paketų („*out-of-the-box*“), skirtų atskiroms vartotojų grupėms, rinkai pateikė karkaso (*framework-oriented*) tipo programinių sprendimų. Karkaso tipo sprendimus sudaro

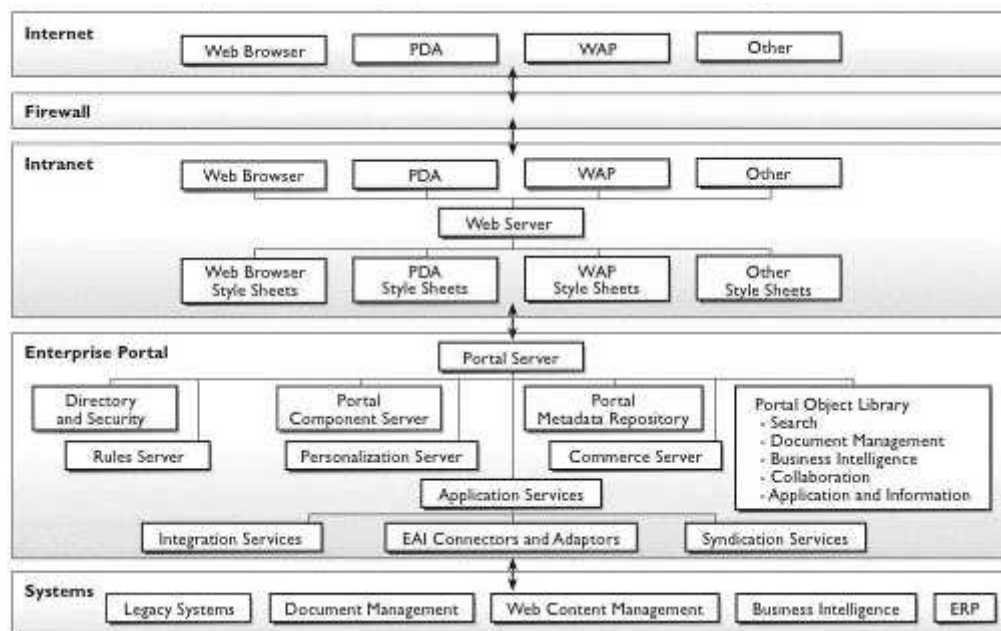
programinių modulių rinkinys ir technologija, kaip iš atskirų modulių sukurti vientisą, vartotojo poreikius atitinkančią informacinę sistemą.

2.3.1.1. Organizacijos žinių portalų techninė architektūra

„Programų sistemų architektūra – tai sistemos struktūra, susidedanti iš programinių komponentų, komponentų matomų arba išmatuojamų savybių, ir ryšių tarp komponentų“[1].

Nesuderinus sistemos techninės architektūros sprendimų ir programinės įrangos architektūrinių sprendimų, šios klaidos gali iškilti tik programinės įrangos diegimo fazėje, kas reikštų didelius klaidų šalinimo kaštus. Tinkamai parinkta architektūra savo ruožtu gali sąlygoti tai, kad kuriama aplikacija lengvai integruosis su esamomis aplikacijomis, nebus dubliuojami veiksmai ir duomenys. Žinomų portalų gamintojai kiekvienas pateikia savo architektūrinį modelį. Šiame skyriuje apžvelgsime apibendrinta portalų architektūros modelį, kuris orientuotas ne į konkretaus portalų gamintojo siūlomus sprendimus, bet į principines technologijas.

Organizacijos žinių valdymo portalų techninę architektūrą sudaro organizacijos tinkle naudojamos technologijos. Šias technologijas tikslinga grupuoti į technologinius sluoksnius.



2.5 pav. Korporacijos žinių portalų informacinių technologijų techninė architektūra [1]

Sluoksniai:

1. Pirmas sluoksnis – Extranetas ir internetas. Šis sluoksnis apibrėžia sąsajas, kurias naudoja klientai ir partneriai. Naudotojo sąsają sudaro skirtingi įrenginiai ir programos: interneto naršyklės,

delniniai kompiuteriai, mobilieji telefonai ir kiti elektroniniai įrenginiai.

Šiame sluoksnyje sprendžiami klausimai:

- Naudotojo sąsaja. Naudotojo sąsajoje turėtų būti numatytos nuorodos į pastaruoju metu lankytas sritis, pasirinktas ekrano išdėstymas, navigaciniai valdymo elementai
- Turinys.
- Turinio pateikimo būdai. Pateikimo būdai priklausomai nuo poreikių, gali būti interneto puslapis, elektroninis laiškas, SMS žinutė ir kitos priemonės.

2. Antras sluoksnis – ugniasienė ir demilitarizuotoji zona. Šiame sluoksnyje realizuojama tinklo apsaugos politika, programinė įranga ir architektūriniai sprendimai, leidžiantys vartotojams naudotis organizacijos resursais. Demilitarizuotoji zona – tai nedidelis kompiuterių tinklas, esantis tarp organizacijos (privataus) tinklo ir išorinio (viešo) tinklo. Būtent šioje aplinkoje bus talpinami internetiniai puslapiai ir kitas turinys, kuriuo galės naudotis išoriniai vartotojai. Šis principas – tai papildoma apsauga nuo įsilaužimų, kadangi demilitarizuotąją zoną pralaužęs vartotojas dar nepateks į vidinį organizacijos tinklą.

Šiame sluoksnyje sprendžiami klausimai:

- Vartotojų autentifikacija
- Ugniasienės techninės ir programinės priemonės
- Interneto apsaugos programinės priemonės
- Tinklo valdymas
- Tinklo apsauga
- Tarpinio serverio (*proxy*) programinės priemonės
- Internetinių aplikacijų valdymas

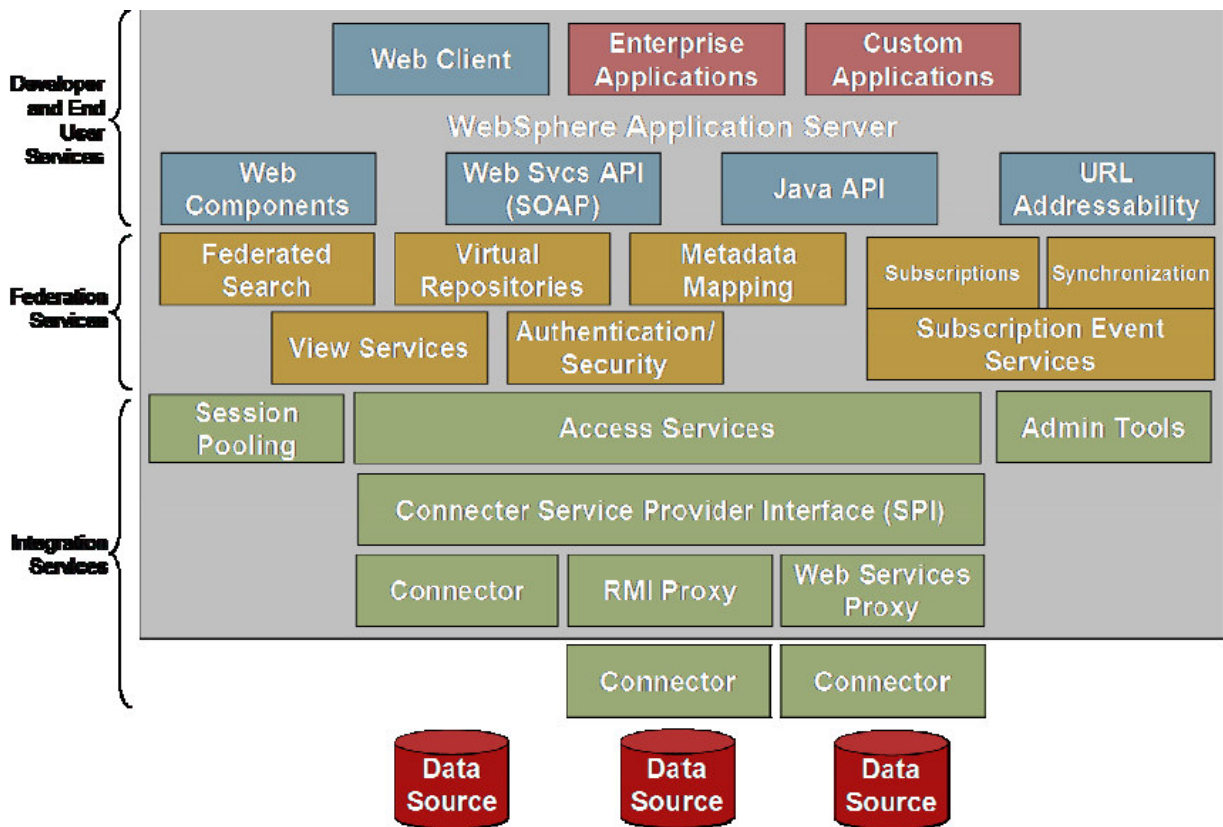
3. Trečias sluoksnis – intranetas. Šį sluoksnį sudaro sąsajos, kuriomis naudojasi organizacijos darbuotojai. Skirtinguose įrenginiuose, duomenys vaizduojami tam įrenginiui specifiniu stiliaus šablonu.
4. Ketvirtas sluoksnis – Organizacijos žinių portalas. Šį sluoksnį sudaro programinės įrangos sprendimai, leidžiantys darbuotojams ir klientams susikurti arba priderinti sąsajas su tinkle esančiais resursais (aplikacijomis, duomenimis, įrengimais ir pan.) asmeniniams poreikiams. Organizacijos žinių valdymo portalų funkcionalumas apima informacijos pateikimą, žinių organizavimo sistemas, paiešką ir indeksavimą, asmeninius nustatymus, roles ir pan. Portalo

modulinė struktūra leidžia atlikti individualius nustatymus, papildyti arba keisti atskirų modulių funkcionalumą neperkuriant visos sistemos.

5. Penktas sluoksnis – taikymo ir adaptavimo sluoksnis. Šis sluoksnis apibrėžia metodus ir įrankius, naudojamus organizacijos sistemų apjungimui ir koordinavimui.
6. Šeštasis sluoksnis – aplikacijos ir sistemos. Šis sluoksnis sudaro organizacijos duomenų saugyklos ir specializuotos aplikacijos (liktinės sistemos, dokumentų tvarkymo sistemos, turinio valdymo sistemos, verslo valdymo sistemos ir pan.)

Sekančiuose skyriuose pateikiami konkrečių portalų gamintojų siūlomi techniniai sprendimai.

2.3.1.2. IBM Organizacijos Informacinis Portalas (OIP)



2.6 pav. IBM WebSphere II portalo architektūra

IBM OIP apibrėžia architektūrinį organizacijos portalo karkasą. Kaip technologinę bazę IBM siūlo produktą, leidžiantį atlikti struktūrizuotų ir nestructūrizuotų duomenų paiešką IBM duomenų šaltiniuose (pvz. DB2), ne IBM tipo reliacinėse duomenų bazėse (pvz. Oracle) ir Lotus Korporacijos sistemose (pvz. Domino, Notes). IBM OIP karkaso pagrindu kuriamos galutiniam vartotojui skirtos programos. Kurdama

OIP karkasą, IBM bendradarbiavo su 33 kompanijom (toliau – partneriais), kuriančiom specializuotus naudotojo sąsajos, asmeninių nustatymų, bendradarbiavimo įrankius ir produktus. Šio bendradarbiavimo rezultate karkasas buvo papildytas sąsajom (API – *Application Programming Interface*) su visų bendradarbiavusių kompanijų sukurtais specializuotais produktais.

IBM apibrėžė karkasą vykdydama strateginį tikslą – padėti įvairių pramonės šakų organizacijoms sėkmingai pradėti elektroninį verslą. Sukurtas sprendimas leidžia pasinaudoti geriausiomis IBM sukurtomis technologijomis, be to, atviros architektūros dėka leidžia sąlyginai lengvai adaptuoti trečiųjų šalių specializuotus produktus.

Karkasas apibrėžia architektūrinius sprendimus, sudarančius bazinį portalų funkcionalumą:

1. Prieigos ir jungtinės paieškos funkcionalumas leidžia peržiūrėti skirtingose struktūrizuotuose ir nestruktūrizuotuose duomenų šaltiniuose paskirstytą informaciją. Duomenų šaltiniai – elektroninis paštas, faksas, dokumentų tvarkymo sistemos, OLAP aplikacijos, ERP sistemos ir kt.
2. Klasifikavimo funkcionalumas leidžia automatiškai pildyti meta duomenų saugyklas iš skirtingų duomenų šaltinių, kas leidžia identifikuoti ryšius tarp informacinių elementų.
3. Profiliavimo funkcionalumas skirtas duomenų apie atskirus darbuotojus ir jų kompetencijas kaupimui.
4. Personifikacijos funkcionalumas leidžia vartotojams adaptuoti portalą pagal jų specifinius poreikius.
5. Bendradarbiavimo funkcionalumas leidžia formuoti specializuotas grupes ir darbo aplinkas. Šio funkcionalumo dėka vartotojai gali keistis žiniomis tiek realiaame laike, tiek ir naudodami bendras duomenų bazes.
6. Aplikacijų integravimo funkcionalumas leidžia realizuoti duomenų lygmens integraciją ir procesų lygmens integraciją, realizuotą pagal J2EE platformoje apibrėžtą jungiklio šabloną.
7. Saugumo ir sistemos valdymo funkcionalumas garantuoja sistemoje esančių duomenų apsaugą, o taip pat leidžia supaprastinti vartotojų identifikavimą (pavyzdžiui, vartotojui užtenka prisijungti vieną kartą, kad jis turėtų prieigą prie įvairiuose duomenų šaltiniuose esančios jam teiktinos informacijos).
8. Unifikuotas portalų vystymo funkcionalumas apima programavimo aplinką, įrankių rinkinius, bendras sąsajas (API), standartines atliktų pakeitimų perkėlimo priemones.

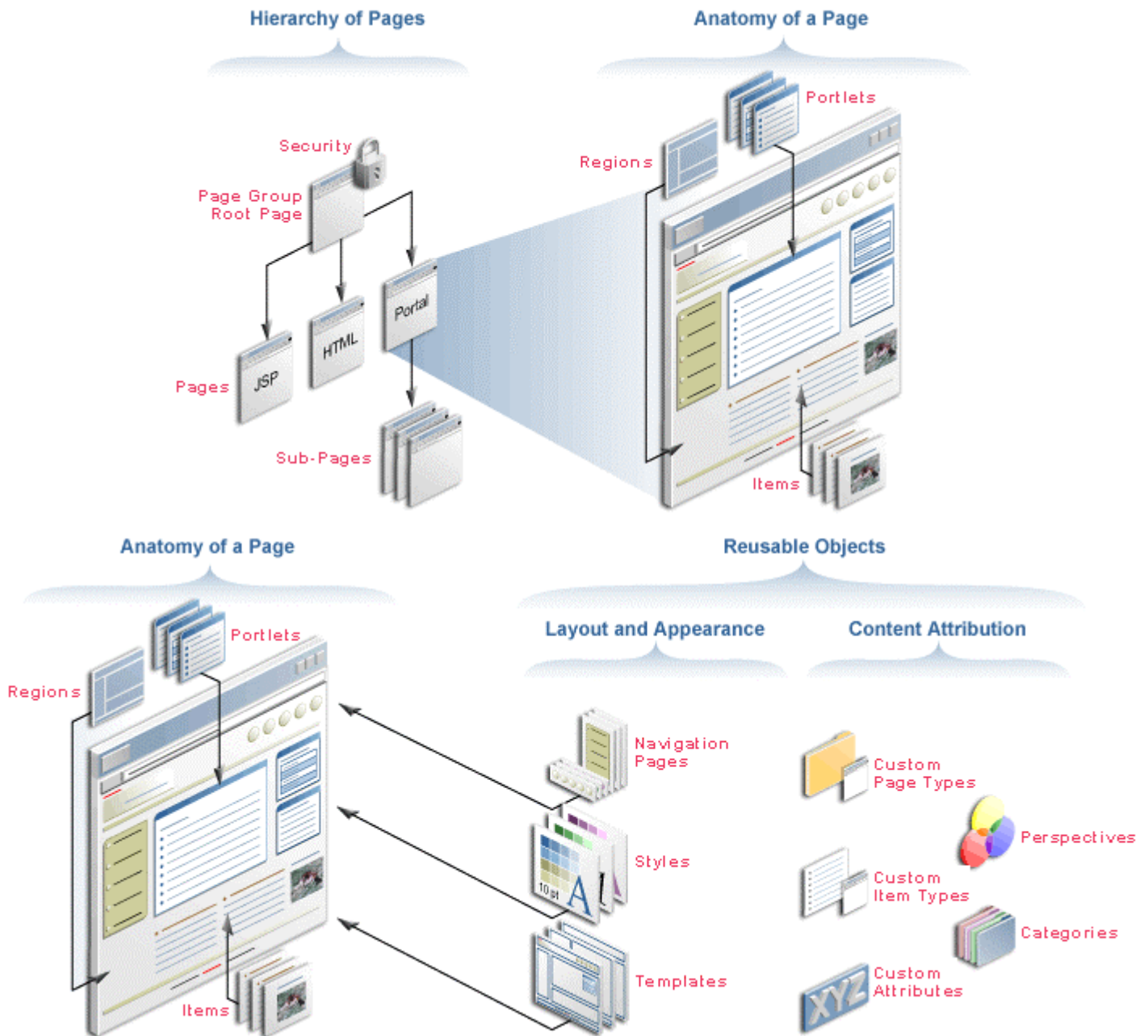
Karkaso architektūra – tai daugiasluoksnė architektūra. Techninė sluoksnių specifikacija:

1. Naudotojo sąsajos sluoksnis. Atviros architektūros principas leidžia kurti naudotojo sąsajas panaudojant skirtingus karkase numatytus įrankius (pvz. IBM Websphere Studio ir KnowledgeX, o taip pat karkaso kūrime dalyvavusių partnerių, tokių kaip Brio, Plumtree, sukurtus produktus).
2. Prieigos ir integravimo sluoksnis. Šį sluoksnį sudaro IBM ir trečių šalių sukurti komponentai (komponentai katalogų/duomenų saugyklų kūrimui, paieškos sistemos, interneto naršyklės ir pan.) prieigos, paieškos, duomenų klasifikavimo, meta duomenų kaupimo, personifikacijos funkcionalumo palaikymui.
3. Informacinių prieigų sluoksnis. Šį sluoksnį sudaro prisijungimo prie daugumos sistemos vidinių ir išorinių informacijos šaltinių komponentai. Sukurti prisijungimo prie reliacinių duomenų bazių (DB2, Oracle, Sybase, Informix, MS SQL Server), nestruktūrizuotų duomenų šaltinių (dokumentų tvarkymo sistemų, tokių kaip IBM EDMSuite, Digital Library, Domino.Doc, Documentum; verslo analitikos (*business intelligence*) sistemų, tokių kaip Business Objects) komponentai. Esant poreikiui, panaudojant atvirą API gali būti sukurti specifiniai prisijungimo komponentai.
4. Aplikacijų sluoksnis. Sluoksnį sudaro ERP/CRM/SCM aplikacijos, integruotos naudojant IBM CommonStore technologiją ir karkaso kūrime dalyvavusių partnerių sukurtas sąsajas. Elektroninės komercijos aplikacijos integruotos naudojant IBM WebSphere technologiją ir karkaso kūrime dalyvavusių partnerių sukurtas sąsajas.
5. Bendradarbiavimo sluoksnis. Sluoksnį sudaro Lotus Development Corp. Bendradarbiavimo ir komandinio darbo (groupware) komponentai (pvz. Notes). Su kitomis aplinkomis bendravimas vyksta per partnerių sukurtus prisijungimo komponentus (connectors)
6. Valdymo sluoksnis. Sluoksnį sudaro konsoliduotos ir unifikuotos standartinių portalų sistemų valdymo funkcijos, apimančios saugumo, naudotojų registravimo ir administravimo, įspėjimų, sistemos našumo ir resursų valdymą.
7. Sistemos vystymo sluoksnis. Sluoksnį sudaro unifikuota, išplėsta vystymo aplinka.

Apibendrinant, IBM portalų karkasas yra pavyzdys, kai programavimo paslaugų teikėjai pasinaudodami sukurta architektūra arba ją išplėsdami pasiūlo rinkai karkaso tipo produktą, kurio pirkėjai ir partneriai gali kurti aukšto lygio sprendimus pasinaudodami karkaso apibrėžtomis rekomendacijomis ir sukomplektuotais įrankiais.

2.3.1.3. Oracle portalų karkasas

Oracle portalų karkasas sukurtas siekiant sustabdyti nesuderintų ir pasenusių internetinių puslapių plitimą ir pagreitinti įmonės portalų kūrimą.



2.7 pav. Oracle portalo struktūra

Oracle portalų karkaso koncepcijos pagrindą sudaro pakartotinio panaudojimo vidiniai arba išoriniai informaciniai komponentai – portletai. Portletai – tai komponentai, talpinantys dažnai naudojamą įmonės

vidaus informaciją (biudžeto atnaujinimai, darbuotojų skaičius, katalogai) ir išorinę informaciją (rinkos rodikliai, orai, naujienos), skirtą konkrečiam taikymui. Portletų rinkiniai saugomi elektroninėse saugyklose, iš kurios kurdamas specifinį portalą, vartotojas tiesiog išsirenka tam tikrą informaciją teikiantį portletą, ir įterpia norimoje vietoje.

Iš technologinės perspektyvos, portletai – tai objektai, sukurti panaudojant standartinius Java API informacijos šaltinius arba kitas programas.

Portletuose, kaip taisyklė, yra naudojamos karkaso infrastruktūros teikiamos funkcijos, tokius kaip vienkartinio prisijungimo galimybė panaudojant LDAP tipo katalogus, personifikuotus duomenų šaltinius, mobilių įrenginių palaikymas, verslo proceso valdymas, bendradarbiavimo funkcionalumas (diskusijos, kalendoriai, įvykiai ir pan.). Komponentai vykdomi aplikacijų serveryje (Oracle atveju – Oracle WebDB serveryje) ir gražina aktualų turinį HTML pavidalu. Portletai kuriami naudojant specializuotą programavimo aplinką – Oracle Portlet SDK.

Oracle aplikacijų serveris WebDB

Pagrindinį vaidmenį Oracle portalų karkase vaidina Oracle aplikacijų serveris WebDB, kuris yra Oracle Internet Application Server (iAS) sudėtinė dalis. WebDB aplikacijų serveris suteikia technologines galimybes, tokias kaip organizacijos duomenų integravimas ir taikymas panaudojant portletus, karkaso infrastruktūrinių funkcijų panaudojimas portletuose ir pan.

Aplikacijų integravimo funkcionalumas karkaso viduje sukurtas panaudojant „hub-adapter“ architektūrinį modelį. Duomenų apsikeitimui su liktinėmis informacinėmis sistemomis ir ERP aplikacijomis naudojamas XML. Gauta informacija portletuose ši apipavidalinama ir paruošiama talpinimui bet kurioje tinklapio vietoje. Duomenų analizės funkcionalumą įgalina oracle „Business intelligence“ ir duomenų saugyklų įrankiai, tokie kaip Oracle Reports, Oracle Discoverer ir Oracle Express. Visi šie produktai yra integruoti Oracle portalų karkase.

Oracle taip pat skiria daug dėmesio trečiųjų šalių dalyvavimui vystant karkasą. Tuo tikslu sukurta Oracle elektroninio verslo portalų partnerių iniciatyvinė bendrija, kurios tikslas – populiarinti Oracle karkasą tarp žinomų integruotų sistemų kūrėjų, turinio valdymo informacinių sistemų ir trečiųjų šalių programinės įrangos kūrėjų.

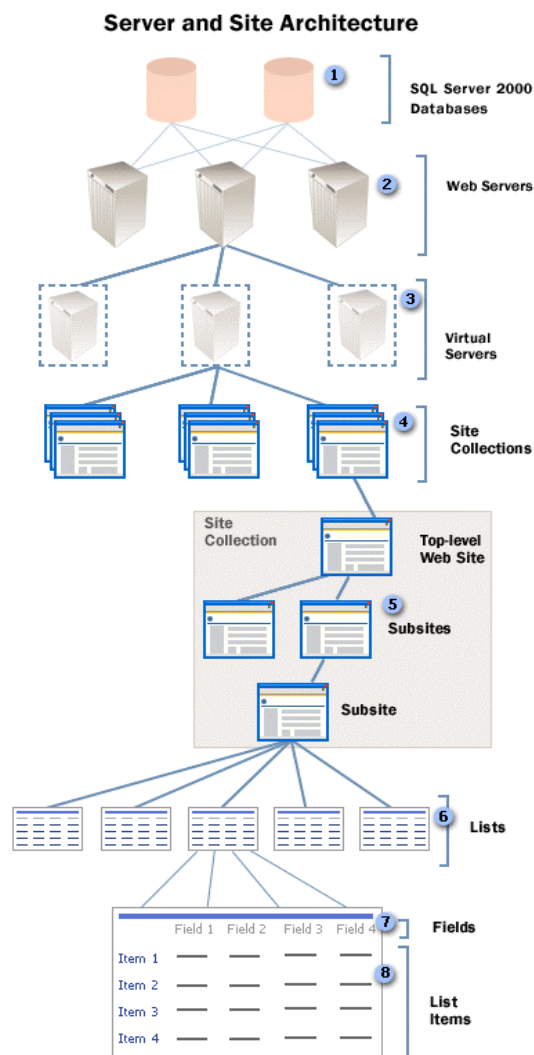
2.3.1.4. Microsoft Share Point

Microsoft taip pat sukūrė portalų karkasą, leidžiantį efektyviai panaudoti Microsoft technologinius sprendimus.

MS SharePoint sudėtiniai komponentai:

1. Vienas ar keli Web serveriai.
2. Viena konfigūracijos duomenų bazė (*configuration*). Konfigūravimo duomenų bazė palaiko atlieka administravimo vaidmenį – nukreipia užklausas į atitinkamas duomenų bazines, reguliuoja Web serverių apkrautumą. Gavus užklausą, šioje bazėje nustatoma kurioje turinio duomenų bazėje yra reikalingi duomenys.
3. Viena turinio duomenų bazė (*content*). Šioje duomenų bazėje saugomas portalo turinys (dokumentai, sąrašai, WebPart savybės, naudotojų duomenys).

Šie komponentai gali būti įdiegti tiek viename kompiuteryje, tiek atskiruose kompiuteriuose.



2.8 pav. SharePoint serverio ir svetainių architektūra

Microsoft Windows SharePoint portalas sudarytas struktūrizuoto hierarchinio objekcinio modelio pagrindu. Šis modelis leidžia nesunkiai pasiekti bet kurį SharePoint svetainės objektą. Hierarchinė organizacija leidžia nesunkiai pasiekti tiek žemesnį, tiek aukštesnį hierarchijos objektą.

Web serveryje gali būti laikoma daug svetainių (*site collection*). Svetainių rinkinys detalizuojamas konkrečiomis svetainėmis (*site*). Svetainė čia suprantama kaip aukščiausio lygmens puslapių loginė sąsaja. Iš čia seka, kad svetainę sudaro susijusių puslapių hierarchija.

Svetainės puslapis logiškai skaidomas į puslapio sritis – regionus. Tose srityse talpinama tiek statinė informacija, tiek dinaminiai sąrašai (*lists*). Sąrašai portale skirti informacijos grupavimui, ir jų tvarkymui skirta daug priemonių. Šalia numatytųjų sąrašų kontaktams, pranešimams, įvykiams, užduotims, nuorodoms, numatyta galimybė susikurti bet kokios struktūros sąrašus, juos pildyti tiek rankiniu būdu, tiek importuoti iš Microsoft Excel lentelių, XML ir žiniatinklio paslaugų (*web service*).

Portalas integruotas su Microsoft Office 2003 paketo programomis. Jame taip pat sukurtos priemonės komandiniam darbui pagerinti – susirinkimų organizavimo priemonės (laiko derinimas, susirinkimo medžiaga, sprendimai, protokolai, dalyviai ir kt.), diskusijų organizavimo priemonės, apklausų organizavimo priemonės, dokumentų ir paveikslėlių bibliotekos, paieškos priemonės tiek svetainių puslapiuose, tiek ir dokumentų bibliotekose esančių dokumentų turinyje. Dokumentai bibliotekoje gali būti versijuojami, realizuotos pasikeitimų sekimo ir informavimo apie pakeitimus priemonės (*alerts*).

SharePoint portalas realizuotas žiniatinklio paslaugų pagrindu, Microsoft .NET platformos aplinkoje. Struktūrą aprašančios klasių bibliotekos gali būti prijungtos prie naujai kuriamos aplikacijos, tuo būdu kuriama aplikaciją galima integruoti portale.

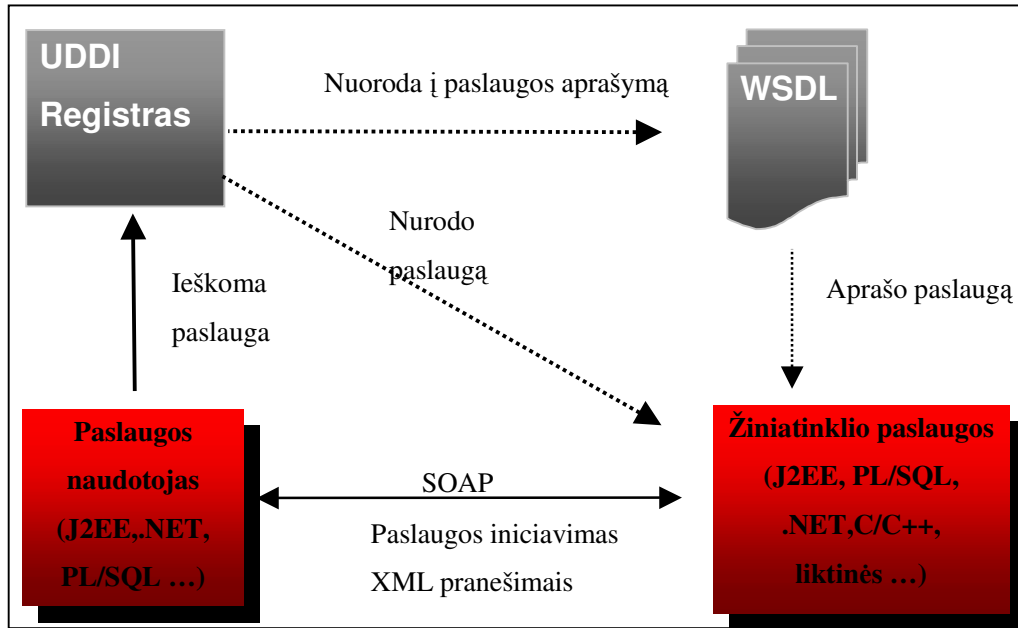
WebPart – tai Microsoft SharePoint portalo aplinkos pakartotino panaudojimo komponentai. Jais svetainės puslapiuose pateikiama specifinė informacija arba paslaugos. WebPart komponentai suprojektuoti taip, kad juos įterpti svetainės puslapyje tiek kaip atskirus komponentus, tiek susieti su kituose komponentuose esančiais duomenimis. Anksčiau minėti sąrašai, bibliotekos – tai standartiniai WebPart technologijos pagrindu realizuoti komponentai. Specifiniams uždaviniams spręsti WebPart komponentus galima programuoti panaudojant SharePoint paketus.

2.3.2. Procesų lygmens integracijos technologijos

2.2.2 skyrelyje aprašyti aplikacijų integravimo principai. Didžiausias integravimo galimybes teikia procesų lygmens aplikacijų integracija.

2.3.2.1. Į paslaugas orientuota architektūra

Teorines procesų lygmens integracijos prielaidas suformavo žiniatinklio paslaugų (*web service*) ir į paslaugas orientuotos architektūrinio (SOA – *Service Oriented Architecture*) šablono atsiradimas. Šablonas pateikia programinių modulių funkcijas paslaugų naudotojų ir paslaugų teikėjų požiūriu.



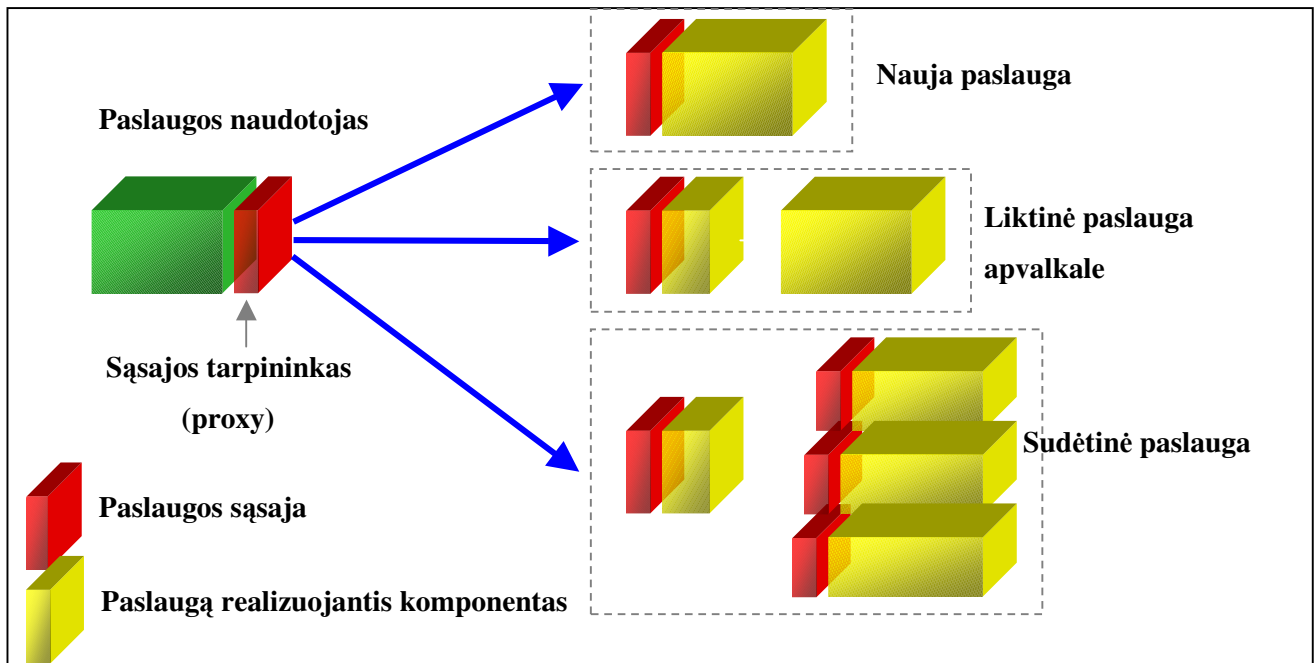
2.9 pav. Žiniatinklio paslaugų (*web services*) dalyviai

Žiniatinklio paslaugų dalyviai

- **Paslaugos naudotojas.** Paslaugos naudotojas gali būti bet kurioje programavimo kalboje aprašytas objektas, galintis siųsti ir priimti XML pranešimus SOAP (*Simple Object Access Protocol*) protokolu.
- **UDDI registras.** Viešos paslaugos paskelbiamos UDDI registruose ir paslaugų naudotojai gali susirasti paslaugą pagal pavadinimą, paslaugos teikėją ar aprašymą. UDDI registras teikia paieškos funkcijas, todėl jei konkrečiu atveju paslaugos yra žinomos, UDDI registras bereikalingas.
- **WSDL (*Web Service Definition Language*).** Tai yra žiniatinklio paslaugų sąsajos aprašymo standartas. Aprašymas pateikiamas XML formatu.
- **Žiniatinklio paslaugos.** Tai yra bet kurioje programavimo kalboje realizuota žiniatinklio paslauga.

Žiniatinklio paslauga – tai gerai apibrėžta funkcija, kuri yra uždara ir nepriklauso nuo kitų paslaugų konteksto.

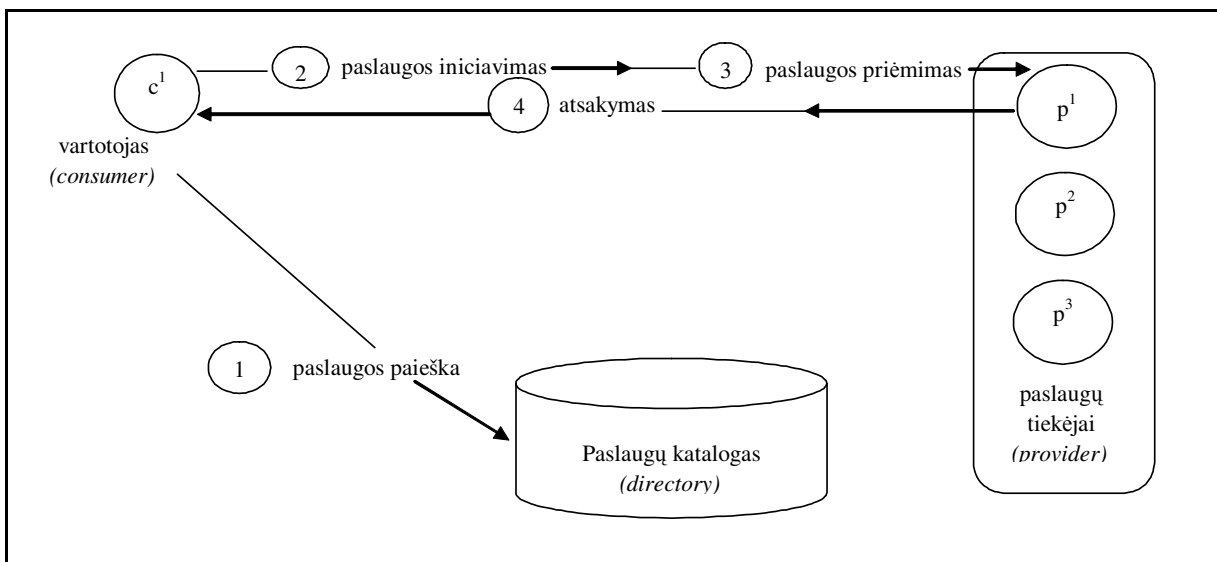
Šis architektūrinis šablonas skirtas „užklašimas – atsakymas“ tipo aplikacijoms. Aplikacijų funkcijos yra išskaidytos į modulius ir pateiktos kaip paslaugos (*services*). Tokios paslaugos turėtų būti mažai tarpusavyje susijusios.



2.10 pav. Paslaugų struktūra ir sąveika

Į paslaugas orientuotas architektūrinis šablonas paslaugą struktūriškai skaido į dvi dedamąsias: paslaugos sąsają ir paslaugą realizuojantį komponentą. Šis principas leidžia panaudojant aplankus (*wrapper*) teikti paslaugas, realizuotas liktiniuose programiniuose moduluose (2.10 pav.).

Architektūrinis šablonas skirtas tokiai aplinkai, kurioje yra daug paslaugų teikėjų ir daug paslaugų naudotojų. Paslaugos panaudojimo seka pavaizduota 2.11 pav. Pirmiausia paslaugų naudotojas susiranda reikiamą paslaugą, tada inicijuoja jos vykdymą. Paslaugų teikėjas priima paslaugos užklausą, suteikia paslaugą ir grąžina atsakymą.



2.11 pav. Paslaugos panaudojimo veiksmų seka

Į žiniatinklio paslaugos (*web service*) sudaro techninę bazę, reikalingą aplikacijų operaciniam suderinamumui (*interoperability*) užtikrinti. Tačiau aukštesnio lygmens valdymas nėra reglamentuotas, t.y. nenurodoma, kurios paslaugos turėtų būti iškviettos, kurie veiksmai įvykdyti ir koks jų eiliškumas. Ši trūkstama grandis užpildoma panaudojant procesų tėkmės (*process flow*) valdymo priemones.

2.3.2.2. BPEL

BPEL (*Business Process Execution Language*) tai paslaugų integravimo technologija, leidžianti iš sinchroninių ir asinchroninių žiniatinklio paslaugų (*web service*) suformuoti veiklos procesą. Standartų ir pramonės konsorciumas OASIS patvirtino BPEL kaip standartą šioje srityje. BPEL yra palaikoma visų didžiųjų programinės įrangos gamintojų (tame tarpe, Oracle, Microsoft, IBM, SAP, Siebel, BEA ir SUN).

BPEL realizuoja tiek sinchroninę, tiek asinchroninę sąveiką, apibrėžia priemones procesų tėkmės valdymui bei kompensuojančiųjų transakcijų valdymui.

BPEL veiklos procesą apibrėžia kaip informacijos srautų valdymą tarp proceso dalyvių. Proceso dalyviai (partneriai) – tai procese dalyvaujantys paslaugų teikėjai. Juos apibrėžiant nurodomas iškviatimo būdas, rolė procese.

Procesas formuojamas iš šių primityvių veiksmų: proceso iniciavimas (*invoke*), atsakymo priėmimas (*receive*), XML dokumentų manipuliacija (*assign*), atsakymas (*reply*), klaidos iššaukimas (*throw*), klaidos apdorojimas (*fault handle*) ir pan.

Proceso grandinės struktūrizavimas atliekamas šiais veiksmiais: eiliškumo apibrėžimas (*sequence*), vykdymo sąlygų apibrėžimas (*switch*), asinchroninių procesų iniciavimas (*flow*), asinchroninių atsakymų apdorojimas, ciklų apibrėžimus ir pan.

BPEL varikliai atskiria procesus, laukiančius asinchroninių atsakymų nuo sinchroninių procesų. Tokiu būdu užtikrinamas patikimumas ir išplėčiamumas.

BPEL Komponentai:

1. **BPEL modeliavimo priemonė** (*designer*). Tai priemonė, leidžianti vizualiai aprašyti veiklos procesą reikiamose vietose prijungiant paslaugas teikiančias aplikacijas. Rezultate suformuojama BPEL programa, kuri bus vykdoma BPEL serveryje. BPEL programa užrašoma XML pavidalu.
2. **BPEL serveris**. Tai aplinka, kurioje vykdoma BPEL programa. BPEL serveris instaliuojamas J2EE platformos aplikacijų serveryje (pavyzdžiui, WebLogic, Oracle AS, JBoss, WebSphere).

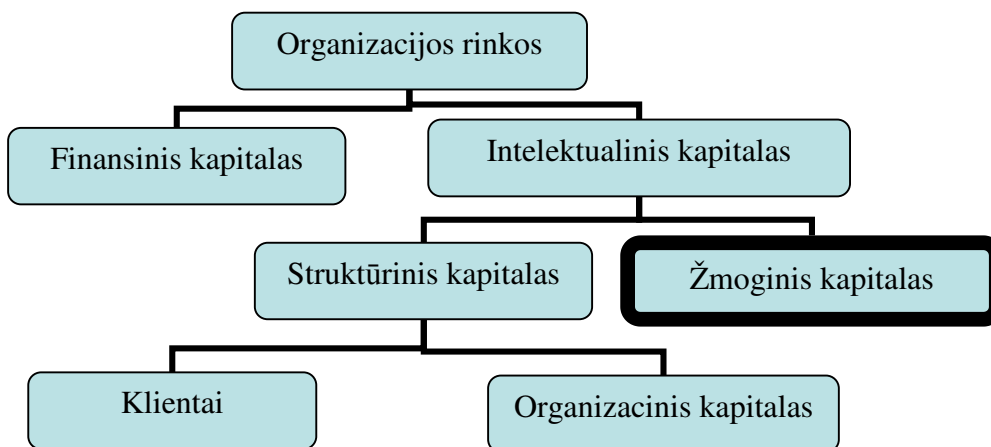
3. **BPEL valdymo pultas** (*console*). Sudaro priemonių rinkinys, skirtas vizualiam procesų vykdymo stebėjimui, valdymui, klaidų taisymui (*debugging*), optimizavimui ir kt.

3. ŽMOGINIO KAPITALO VALDYMO INFORMACINĖS SISTEMOS PROJEKTINIAI SPRENDIMAI

Sukurtas produktas – Žmoginio kapitalo valdymo informacinė sistema (kodinis pavadinimas HCMIS). Tai produktas, skirtas įmonei, kurios viena iš veiklos sferų – projektų vykdymas. Konkretus taikymas atliktas organizacijoje, vykdančioje programinės įrangos kūrimo projektus, tačiau visa sistema paremta dinamiškai keičiamais klasifikatoriais, kuriuos užpildžius kitai taikomajai sričiai, sistema pilnai tiktų.

3.1. Sistemos apibūdinimas

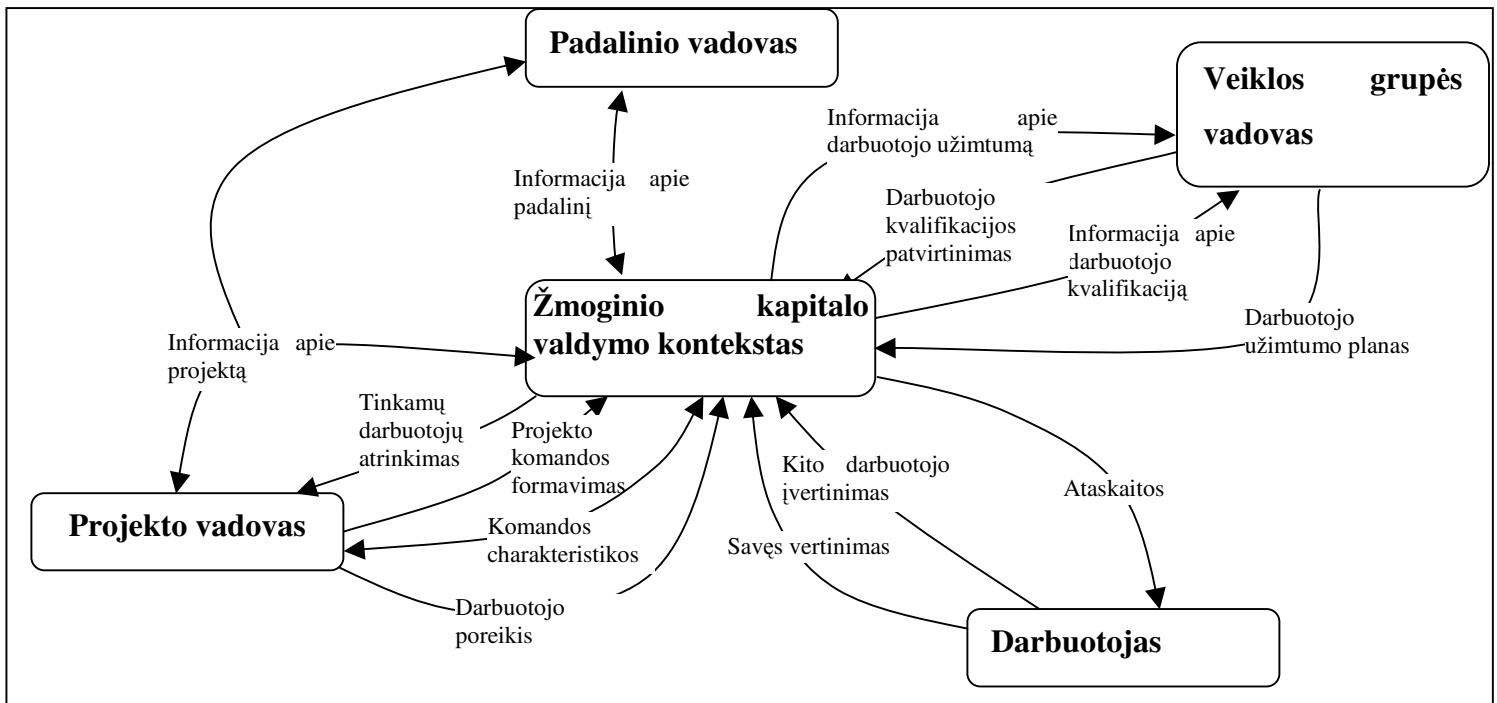
Produktas – žmoginio kapitalo valdymo informacinė sistema (kodinis pavadinimas HCMIS) – orientuotas į organizacijos turimų žmoginių išteklių valdymo problemų sprendimą. Žmoginio kapitalo indėlis formuojant visos organizacijos rinkos vertę pavaizduotas 3.1 pav.



3.1 pav. Organizacijos rinkos vertės struktūra

Sukurtas produktas – tai įrankis, padėsiantis kaupti duomenis apie darbuotojų žinias, sugebėjimus, asmenines savybes, patirtį ir kitas savybes leisiančias efektyviau naudoti organizacijos žmoginius išteklius sprendžiant kasdienines organizacines problemas. Sistemoje esantys duomenys panaudojami darbuotojų kvalifikacijos pokyčių stebėjimui, reikiamos kvalifikacijos darbuotojo paieškai naujam projektui, sudarant komandas atsižvelgiama į darbuotojų tarpusavio vertinimus.

HCMIS konteksto diagramoje (3.2 pav.) pavaizduoti esminiai sistemos naudotojai ir duomenų srautai tarp naudotojų ir sistemos.



3.2 pav. HCMIS konteksto diagrama

Organizacijos žinių valdymo informacinės sistemos:

- Ryšių su klientais valdymo sistema
- Dokumentų tvarkymo sistema
- Buhalterinės apskaitos sistema
- Darbo laiko apskaitos sistema
- Žmoginio kapitalo valdymo informacinė sistema (HCMIS)
- Turinio valdymo sistema
- Kitos specializuotos sistemos (Klaidų registravimo, Versijų valdymo ir pan.)

3.2. Sistemos architektūra

Sistema realizuota trijų sluoksnių architektūra.

Duomenų bazės sluoksnis

Šiame sluoksnyje saugojami sistemos duomenys reliaciniame pavidale. HCMIS sistemoje naudojama SQL Server reliacinė duomenų bazių valdymo sistema. Duomenų tvarkymas atliekamas naudojant duomenų bazių procedūras (*stored procedures*). Tiesioginis lentelių duomenų manipuliavimas draudžiamas.

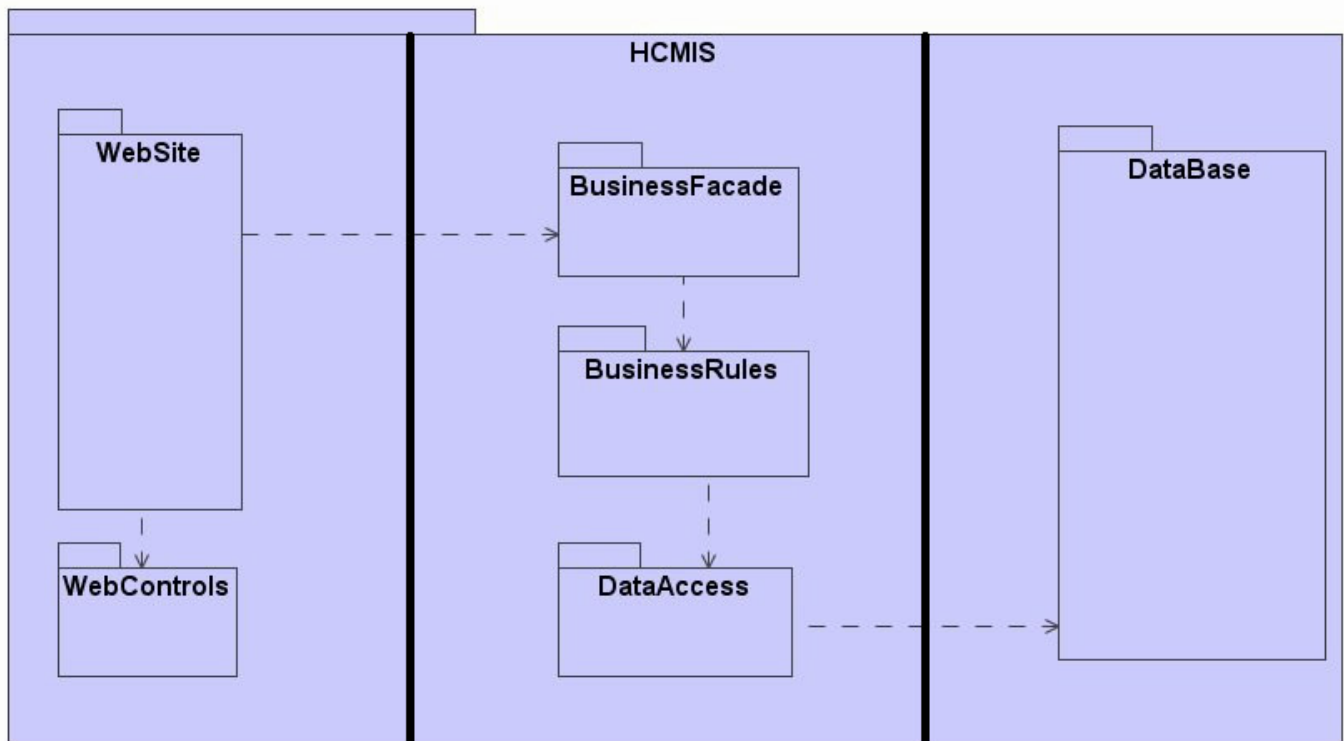
Veikla organizuojama apie tris pagrindinius objektus – organizaciją (lentelių pavadinimai prasideda „org_“), darbuotoją (lentelių pavadinimai prasideda „emp_“ ir veiklą (lentelių pavadinimai prasideda „act“).

Veiklos taisyklių sluoksnis

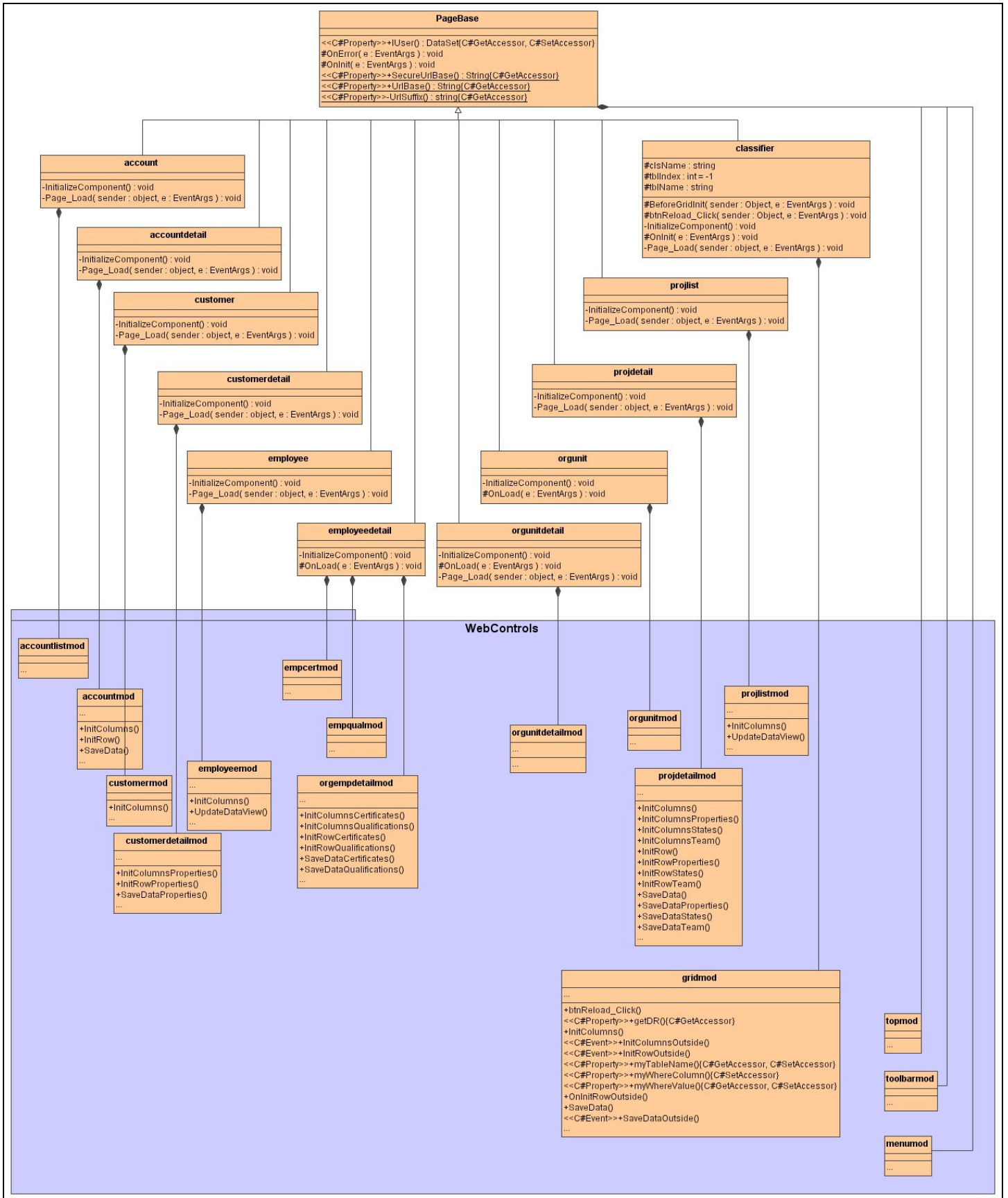
Veiklos taisyklių sluoksnį sudaro trys paketai: BusinessFacade (detalizuota klasių diagrama pavaizduota 3.6 pav.), DataAccess (3.8 pav.) ir BusinessRules. DataAccess paketą sudaro duomenų manipuliavimą užtikrinančios klasės. Tai yra vienintelis paketas, turintis ryšį su duomenų baze. BusinessFacade paketas skirtas atskirti vaizdavimo ir veiklos procesų komponentams. Naudotojo sąsajos komponentai tiesiogiai sąveikauja tik su BusinessFacade paketu. BusinessRules paketo klasėse realizuota veiklos logika, t. y. apibrėžtas duomenų užkrovimo eiliškumas, taisyklės ir kt.

Naudotojo sąsajos sluoksnis

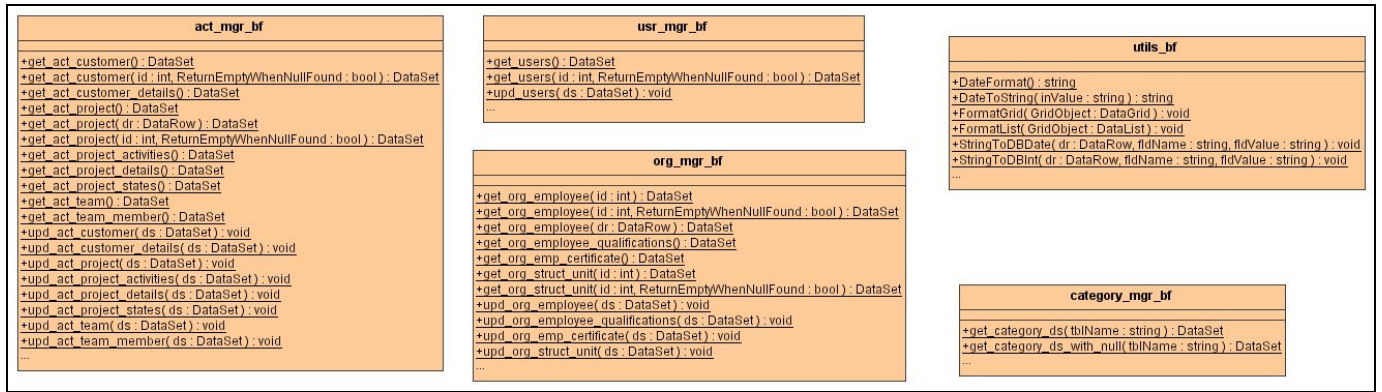
Sluoksnį sudaro du paketai: WebSite (detalizuota klasių diagrama pavaizduota 3.4 pav.) ir WebControls (detalizuota klasių diagrama pavaizduota 3.7 pav.). WebSite paketą sudaro galutiniam vartotojui matomi puslapiai. WebControls paketą sudaro specifiniai, pakartotino panaudojimo valdymo elementai.



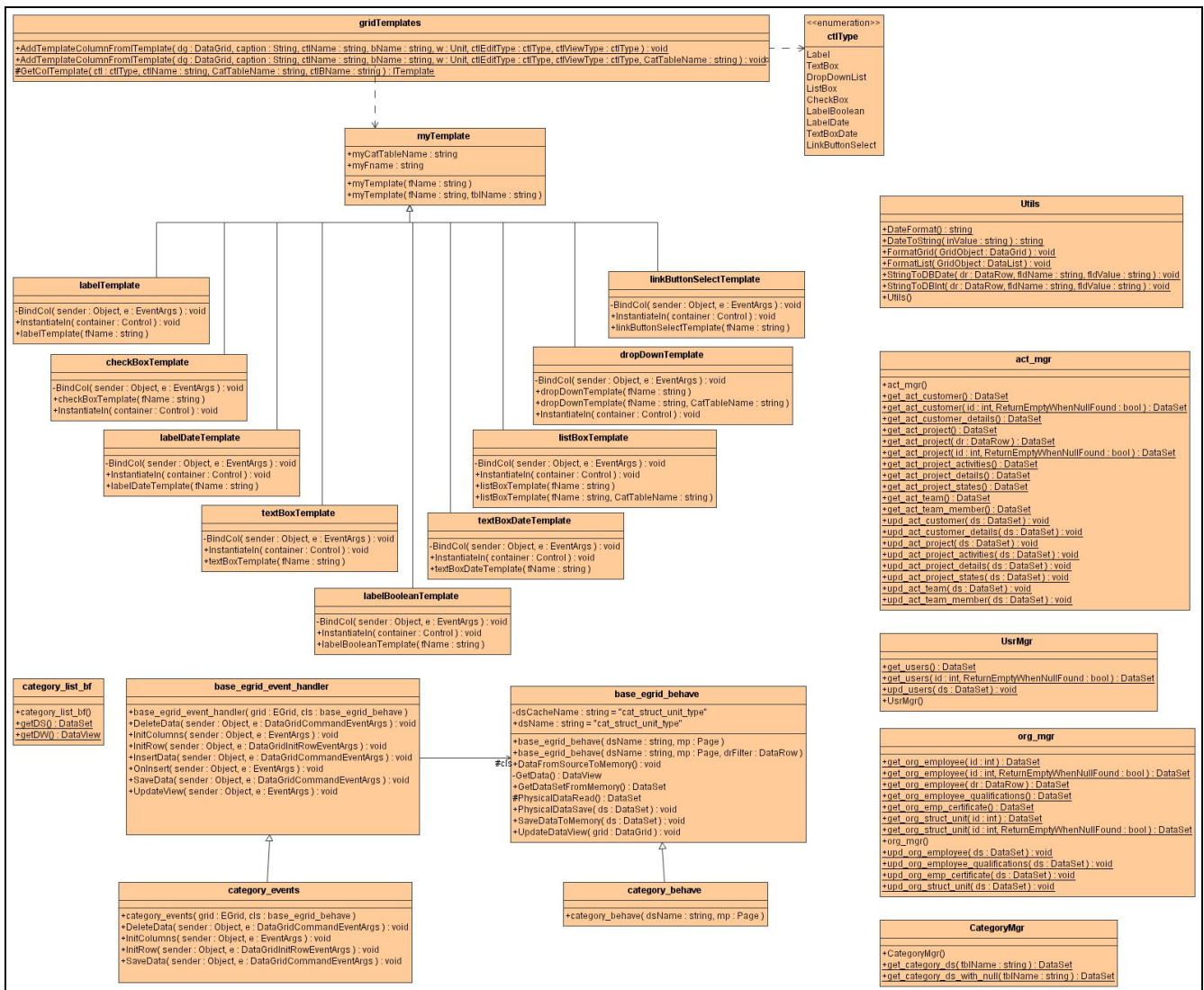
3.3 pav. HCMIS paketų diagrama



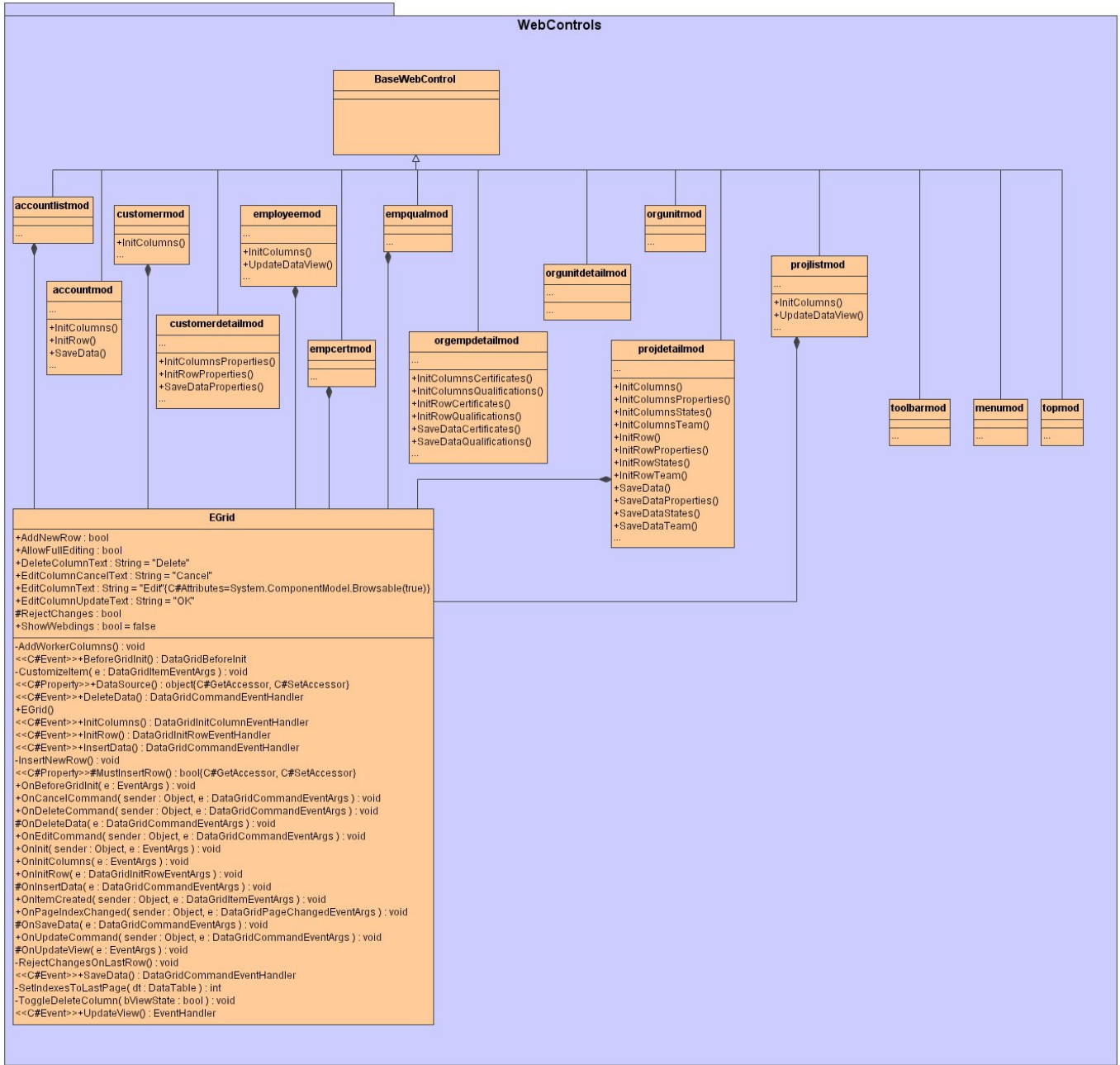
3.4 pav. WebSite paketo klasių diagrama



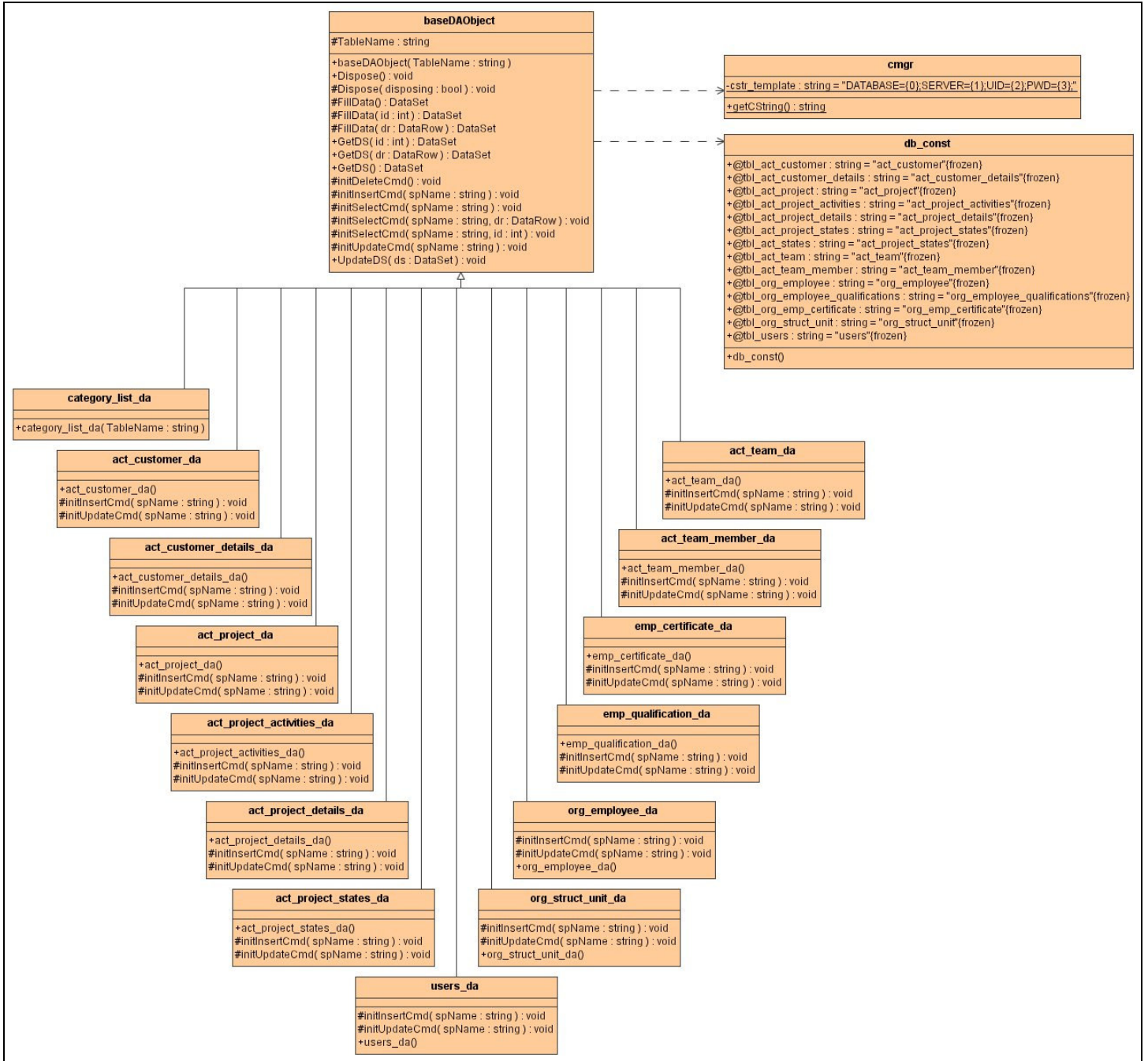
3.5 pav. BusinessFacade paketo klasių diagrama



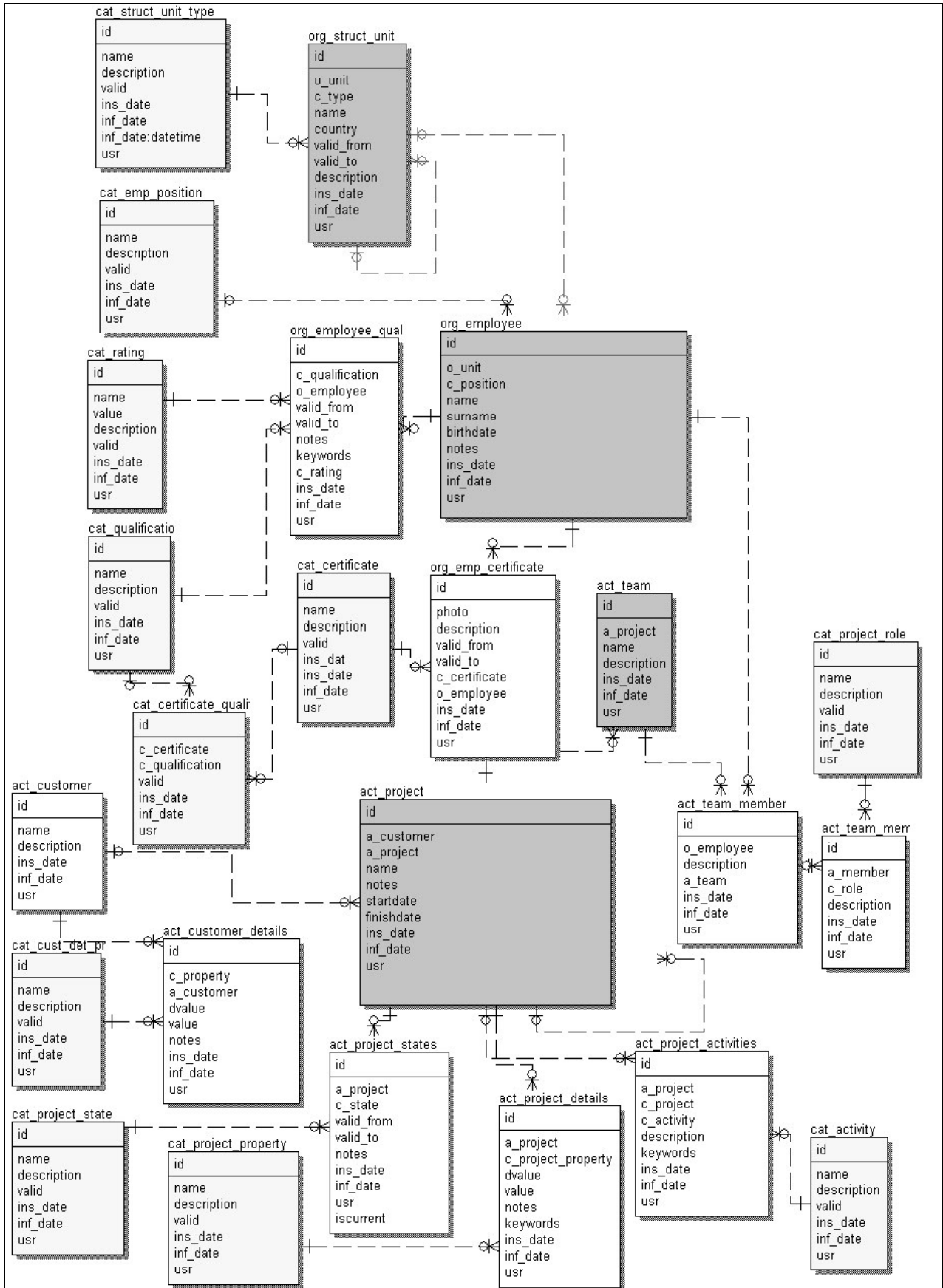
3.6 pav. BusinessRules paketo klasių diagrama



3.7 pav. WebControls paketo klasių diagrama



3.8 pav. DataAccess paketo klasių diagrama



3.9 pav. HCMIS DB schema

4. ŽMOGINIO KAPITALO VALDYMO INFORMACINĖS SISTEMOS PROJEKTINIO SPRENDIMO TYRIMAS

HCMIS sistema sukurta organizacijos žmoniškųjų išteklių informacijos kaupimui. Šiame skyriuje apžvelgiami techniniai sprendimai ir patobulinimai.

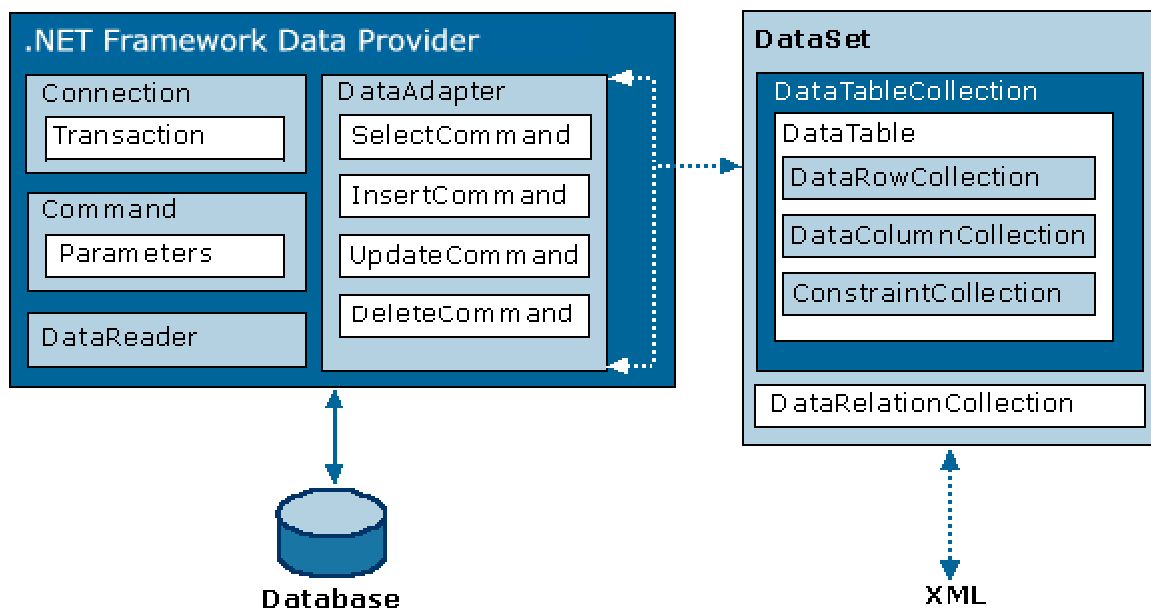
4.1 lentelė Techninė aplinka

Serverio operacinė sistema	Microsoft Windows 2003 Server
DB Serveris	MS SQL 2000
Web Serveris	Microsoft Internet Information Server 6.0
Aplinkos karkasas (Framework)	Microsoft .NET Framework 1.1
Technologijos	.NET, C#, CSS, HTML

4.1. Technologiniai sprendimai

Duomenų prieiga(Data Access)

Duomenų prieiga atliekama panaudojant Microsoft ADO.NET technologiją.



4.1 pav. ADO.NET duomenų prieigos architektūra [8]

Duomenų nešėjai – DataSet tipo objektai. Ši struktūra leidžia atkartoti duomenų bazės objektų vaizdą atminties kintamuosiuose. Šio tipo objektams numatytos standartinės funkcijos vaizdavimui XML

pavidalu, todėl garantuojamas paprastas duomenų apsikeitimas su vidiniais ir išoriniais sistemos komponentais.

Duomenų prieiga vykdoma per baseDAObject klasės išvestines klases. (baseDAObject klasė atitinka .NET Framework Data Provider reikalavimus). Visi standartiniai duomenų prieigos veiksmai (prisijungimas, struktūros ir pan.) realizuoti baseDAObject klasėje, vieninteliai veiksmai, kurie aprašomi atskirai – tai duomenų bazės procedūrų sąsaja su duomenų nešėju duomenų užkrovimo, įterpimo, atnaujinimo ir ištrynimo metu (3.8 pav. Matome, kad iš baseDAObject išvestinėse klasėse perdengti metodai yra tik įterpimo ir atnaujinimo veiksmų paruošimo procedūros).

Naudotojo sąsaja

Naudotojo sąsajos formavimui panaudoti ASP.NET specializuoti naudotojo sąsajos valdymo komponentai (*Custom Web Controls*).

Iš standartinio .NET DataGrid komponento, sukurtas specialus komponentas EGrid duomenų lentelių vaizdavimui ir tvarkymui (vaizdinis fragmentas pavaizduotas 4.2 pav. ir EGrid klasė 3.7 pav. WebControls paketo klasių diagramoje)

State	Valid from	Valid to	Current	Notes		
Elaboration	2004.01.02		False		Select	Edit
Construction	2004.05.01		<input type="checkbox"/>		Select	OK Cancel
Transition	2004.09.01		True		Select	Edit
1						

4.2 pav. Duomenų lentelės vaizdavimo ir tvarkymo komponentas (EGrid)

EGrid komponento eilutės gali būti peržiūros arba redagavimo režime. Kiekvienas stulpelis gali būti susietas su skirtingu valdymo elementu – žyme (*label*), žymimuoju langeliu (*checkbox*), tekstiniu lauku (*input box*), reikšmių sąrašu (*combo box*).

Veiksmai su duomenimis atkartojami vidinėje DataSet tipo klasėje, kuri vėliau panaudojant atitinkamą DataAccess paketo klasę, panaudojama vykdant atitinkamus pakeitimus ir duomenų bazėje. EGrid komponentas realizuoja sąrašo vaizdavimą puslapiais.

Naudotojo sąsaja suformuota iš atskirų, pakartotinio panaudojimo komponentų (*Web Controls*).

Duomenų modelis

Projektuojant duomenų modelį buvo keliami du uždaviniai:

1. Sistemoje kaupiamų duomenų lankstumas.
2. Aukštas duomenų struktūrizacijos lygis.

Norint įvykdyti pirmą uždavinį, duomenų struktūroje galima numatyti daug laisvo teksto laukų, ir juose leisti įvesti bet kokio pobūdžio informaciją. Tačiau šitoks sprendimas prieštarauja antrajam uždaviniui. Šios problemos sprendimui, duomenų struktūroje buvo numatytas visas rinkinys klasifikatorių

(3.9 pav. lentelės, kurių pavadinimai prasideda “cat_”), kuriais apsprendžiamas informacijos pobūdis. Išskirti veiklos objekto informaciniai aspektai (pvz. darbuotojo kvalifikacija, sertifikatai, savybės) ir sukurti kiekvieno aspekto klasifikatoriai ir tarpinės lentelės (3.9 pav. lentelės, kurių pavadinimai prasideda), susiejančios konkretų veiklos objektą ir jo informacinius aspektus.

Tiesioginių veiksmų su duomenų lentelėmis ir vaizdais vykdymas uždraustas. Visi duomenų manipuliavimo veiksmai atliekami tik per išsaugotas duomenų bazės procedūras. Ši nuostata leidžia centralizuoti loginį duomenų organizavimą ir pagerinti integralumą.

Naudotojo sąsajos patobulinimai

Lankstus duomenų modelis leidžia išplėsti projektų, darbuotojų ir kitų veiklos objektų savybių rinkinį. 4.3 pav. matome, kaip pildomas klientų savybių sąrašas.

Customer properties					
Property name	Property value	Notes	Keywords		
E-mail	info@vmi.lt			Select	Edit
Phones	68444456			Select	OK Cancel
Name of enterprise					
Code of the enterprise					
E-mail					
Phones					
Faxes					
Contact person					
Country					

Reload

4.3 pav. Kliento savybių įvedimas

Tačiau augant savybių skaičiui, įvedant naują klientą reikia „atsiminti“, kokias savybes pildyti, be to, išsirinkimas iš sąrašo nėra patogus.

Tokiems atvejams sistemoje įvesti savybių šablonai. Šablone įvedamas savybių rinkinys, ir nurodomas vaizdavimo eiliškumas ir būtinumo požymis. Po to, kai sukuriamas šablonas, įvedus pirkėją, jam užpildomas savybių sąrašas iš šablono (4.5 pav.), ir lieka suvesti jau konkrečias reikšmes lentelėje.

Customer properties template		
No	Name	
1	Name of enterprise	✓
2	Code of the enterprise	✓
3	E-mail	✓
4	Phones	
5	Faxes	✓
6	Contact person	

4.4 pav. Kliento savybių šablonas

Customer properties						
Property name	Property value	Notes	Keywords			
Name of enterprise				Select	Edit	Delete
Code of the enterprise				Select	Edit	Delete
E-mail				Select	Edit	Delete
Phones				Select	Edit	Delete
Faxes				Select	Edit	Delete
1						
Add new		Reload				

4.5 pav. Kliento savybių įvedimas pritaikius šabloną

4.2. Integracija

Integracijos galimybės

HCMIS sistema realizuota Microsoft.NET technologijomis. Veiklos taisyklių sluoksnį sudaro .NET aplinkos reikalaujantys komponentai (Facade.dll, Rules.dll, DataAccess.dll). Esant poreikiui šiuos komponentus panaudoti kitose .NET pagrindu kurtose sistemose, užtenka įtraukti nuorodas į šiuos komponentus. Taigi, su kitomis, ne .NET tipo aplikacijomis, integracija galima tik duomenų lygmenyje.

Rules.dll paketas realizuotas WebService technologijos pagrindu, kas išplečia integravimo galimybes. WSDL leidžia automatizuoti klasių - adapterių kūrimą. O tais atvejais, kai objekcinio programavimo principai neįmanomi, galima sąsają galima realizuoti panaudojant XML ir HTTP.

Integracija portale SharePoint

Naudotojo sąsaja sudaryta iš atskirų naudotojo sąsajos komponentų (WebControls). SharePoint portalo technologija taip pat numato modulinę puslapių kūrimo technologiją (WebParts technologija). Kuriant WebParts komponentus, tiesiog galima sukurti išvestines klases iš esamų WebControls tipo klasių, ir realizuoti specifinius poreikius.

Procesų lygmens integracija

Veiklos funkcijose realizuota žiniatinklio paslaugų sąsaja (web service) leidžia integruoti sistemą tiek duomenų lygmenyje, tiek ir panaudojant veiklos procesų integravimo variklius.

5. IŠVADOS

1. Norint priimti žiniomis grįstus sprendimus, būtina kompiuterizuoti veiklos procesuose naudojamus duomenis, informaciją ir žinias. Didelėje organizacijoje veiklos procesų kiekis ir įvairovė gali būti didelė, todėl kompiuterizacijos procesas dažniausiai yra palaipsninis, o tai lemia nemažas specializuotų programų apimtis.
2. Specializuotose programose sukaupta informacija yra išsibarsčiusi, todėl jos perdavimas ir panaudojimas priimant operatyvius bei strateginius valdymo sprendimo sprendimus, yra sudėtingas. Taip atsiranda integracijos technologijų ir priemonių poreikis.
3. Galima išskirti tris pagrindinius integracijos principus, arba lygmenis: duomenų integracija, pranešimų integracija bei procesų integracija. Integracijos principo pasirinkimą lemia specifiniai organizacijos poreikiai bei finansinės įmonės galimybės.
4. Paprasčiausias integracijos principas – integracija duomenų lygmenyje, siūlomas taikyti organizacijai, naudojančiai vidutinio sudėtingumo aplikacijas bei disponuojančiai ribotais finansiniais ištekliais.
5. Sudėtingesnis – pranešimų lygmens integracijos principas siūlomas taikyti nedidelėms organizacijoms, kurių veiklos procesų kiekis nėra didelis, arba veiklos procesai nėra aiškiai apibrėžti. Pranešimų lygmens integracija gali būti laikinas sprendimas prieš pereinant į procesų lygmens integraciją, kadangi integracijos perėjimas iš pranešimų į procesų integracijos lygmenį yra pakankamai paprastas, t.y. nereikalauja architektūrinių komponento pakeitimų.
6. Sudėtingiausia - procesų lygmens integracija reikalauja griežtai apibrėžtų organizacijos veiklos procesų. Tuomet pritaikius į paslaugas orientuotą architektūrinį šabloną (SOA – Service Oriented Architecture), kompiuterizuotų veiklos procesų koordinavimas atliekamas panaudojant veiklos procesų vykdymo variklius. Veiklos procesus tikslinga aprašyti BPEL (Business Process Execution Language) kalba, kadangi standartų ir pramonės konsorciumas OASIS yra patvirtinęs reikiamus standartus šioje srityje. BPEL kalbą palaiko visų didžiųjų programinės įrangos gamintojų (tame tarpe, Oracle, Microsoft, IBM, SAP, Siebel, BEA ir SUN) aplikacijų serveriai.
7. Informacijos integravimo funkcionalumą teikia žinių portalai. Kuriant portalus, tikslinga pasinaudoti didžiųjų programinės įrangos gamintojų (Microsoft, IBM, Oracle) sukurtais karkaso tipo portalais. Juose sukurtas infrastruktūrinių funkcijų rinkinys leidžia apjungti skirtingus duomenų šaltinius ir sistemas bei pateikti vientisą, personalizuotą informaciją.

8. Sukurta ir projektinėje darbo dalyje aprašyta žmoginio kapitalo valdymo sistema skirta kaupti projektus vykdančios organizacijos veiklos duomenis: duomenis apie įmonės struktūrą, darbuotojus, jų kvalifikaciją, vykdomus projektus, darbuotojų tarpusavio vertinimus. Naudotojo sąsaja sudaryta iš atskirų informacinių modulių.
9. Projektinės dalies pabaigoje buvo išdėstytos sukurtos žmoginio kapitalo valdymo sistemos integracijos galimybės. Sukurtus modulius nesudėtinga integruoti Microsoft SharePoint portale ir toliau vystyti portalo SharePoint portalo bazėje. Veiklos funkcijose realizuota žiniatinklio paslaugų sąsaja (web service) leidžia integruoti sistemą tiek duomenų lygmenyje, tiek ir panaudojant veiklos procesų integravimo variklius.
10. Projektinės darbo dalies rezultatai nurodo galimas tolimesnių tyrimų kryptis:

Pranešimų ir procesų lygmens integracijos grįstos pranešimų apsikeitimais tarp skirtingų komponentų. Tačiau skirtinguose komponentuose gali būti naudojami ne tik skirtingi pranešimų formatai, bet ir pačių pranešimų turinys. Todėl tikslinga tirti priemones, užtikrinančias ne tik pranešimų struktūros transformacijas, bet ir pranešimų turinio transformacijas remiantis organizacijoje sukauptomis žiniomis. Tai leistų aprašyti žinių kaupimo procesus, kurie dažnai neturi tokių aiškių apibrėžimų, kokius turi veiklos procesai.

LITERATŪRA

1. BASS L., CLEMENTS P., KAZMAN R. Software Architecture in Practice, Second Edition. MA: Addison-Wesley, 2003. 560 p. ISBN 0-321-15495-9.
2. CeBASE. Prieiga per internetą: <http://www.cebase.org/www/frames.html?/www/researchActivities/COTS/integrationmodels.html>.
3. COLLINS H. Enterprise knowledge portals : next generation portal solutions for dynamic information access, better decision making, and maximum results. USA : AMACOM, 2003. 448 p. ISBN 0-8144-0708-0.
4. DAVYDOV M. M. Corporate Portals and e-Business integration, USA : McGraw-Hill, 2001. 305 p. ISBN 0-07-138279-8.
5. Microsoft Corporation, (2003). .NET Framework SDK, SharePoint Products and Technologies (2003). Server and Site Architecture Overview. Prieiga per internetą: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/tsovSiteArchitecture_SV01074499.asp?frame=true >.
6. Microsoft Corporation, (2003). Architectural Overview of Windows SharePoint Services. Prieiga per internetą: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odc_SP2003_ta/html/ODC_WSSArchitecture.asp>.
7. Microsoft Corporation, (2003). .NET Framework SDK, SharePoint Products and Technologies (2003). Web Parts Overview. Prieiga per internetą: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/smpxWebPartsOverview_SV01071436.asp?frame=true>.
8. Microsoft Corporation, (2003). .NET Framework Developer's Guide, ADO.NET Architecture. Prieiga per internetą: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconadonetarchitecture.asp>>.
9. Oracle Corporation, (2003). Oracle® Application Server Portal. User's Guide. Prieiga per internetą: http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/DOCTEAM/902DOC/904_DOC_BUILDING_PORTLETS/B13922_01.PDF>.
10. PERKS C., BEVERIDGE T. Guide to enterprise IT architecture. New York : Springer-Verlag, 2003. 473 p. ISBN 0-387-95132-6.

11. ROEDNER D.. The Next Wave of Integration Platforms. EAI Journal, September 2002.
12. THIERAUF R. J. Effective Business Intelligence Systems, London: Quorum books, 2001. 389 p. ISBN 1-56720-370-1.
13. THIERAUF R. J. Knowledge management systems for business, Westport: Quorum books, 1999. 366 p. ISBN 1-56720-218-7.
14. YAKIMOWITCH, BIEMAN, BASILI. Software Architecture Classification for Estimating the Costs of COTS Integration. icse 99. Prieiga per internetą: <http://sunset.usc.edu/icse99/>

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Santrumpos

Santrumpa	Paaškinimas
API	Application Programming Interface
BPEL	Bussiness Process Execution Language
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Common off the shelf
DCOM	Distributed Component Object Model
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
GUI	Graphic User Interface
JDBC	Java Database Connectivity
ODBC	Open Database Connectivity
OIP	Organizacijos informacinis portalas
OLAP	<i>On-Line Analytical Processing</i>
OLE	Object Linking and Embe
WSDL	Web Service Definition Language
XML	Extensible Markup Language

Terminai

Terminas	Paaškinimas
Baltos dėžės pakartotinis panaudojimas	Programinio produkto adaptavimas, kai adaptuojamas komponentas modifikuojamas sukuriant integravimui reikalingas sąsajas
Komponento aplankas (<i>wrapper</i>)	Komponentas, kuris apgaubia integruojamą komponentą, ir per kurį vyksta apsikeitimas pranešimais. Naudojamas kai integruojamas komponentas neturi integracijai reikalingos sąsajos (<i>interface</i>)
Korporacijos aplikacija	Tai aplikacija, gauta sujungus visas korporacijos veiklos procesus kompiuterizuojančias aplikacijas.
Korporacijos aplikacijų integracija	Terminas, apimantis planus, metodus ir įrankius, skirtas korporacijos viduje esančioms aplikacijoms modernizuoti, sujungti ir koordinuoti.

Pernešamumas	Programinių produktų savybė, susijusi su panaudojimu skirtingose aplinkose (pvz. Operacinėse sistemose)
Pilkos dėžės pakartotinis panaudojimas	Programinio produkto adaptavimas keičiant iš anksto numatytų parametrų reikšmes
Portalas	Svetainė, pasižyminti bendro naudojimo paslaugomis (paieška, personifikacija, naujienos, apklausos ir pan.), dažnai naudojama įvairiuose duomenų šaltiniuose esančių duomenų suliejimui.
Portletas	Oracle Portalo komponentai, talpinantys konkrečios paskirties informaciją arba funkcionalumą.
Tarpinė programinė įranga (<i>glueware</i>)	Programinė įranga, įgalinanti komponentų, turinčių skirtingas sąsajas arba veikiančių skirtingose aplinkose bendravimą (pvz. Techninės įrangos tvarkyklės (<i>driver</i>)).
WebPart	SharePoint portalo komponentai, talpinantys konkrečios paskirties informaciją arba funkcionalumą.
Žiniatinklis	Angliško termino web lietuviškas atitikmuo

1 PRIEDAS. BPEL TECHNOLOGIJŲ SAŠAJA

