

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Rolandas Krištapaitis

**Įvairių XML dokumentų migravimo ir analizės
programinė įranga**

Magistro darbas

Darbo vadovas

Doc. dr. Eimutis Karčiauskas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Rolandas Krištapaitis

**Įvairių XML dokumentų migravimo ir analizės
programinė įranga**

Magistro darbas

Recenzentas

doc. dr. Aleksas Riškus
2010-05-27

Vadovas

doc. dr. Eimutis Karčiauskas
2010-05-27

Atliko

IFM-4/2 gr. stud.
Rolandas Krištapaitis
2010-05-27

Kaunas, 2010

TURINYS

Įvadas.....	1
1.1. Darbo tikslai ir rašymo aplinkybės.....	1
1.2. XML dokumentų valdymo programinė įranga.....	2
1.3. Dokumento turinys.....	3
2. XML dokumentai, standartai ir jų problemos.....	4
2.1. Standartų ideologija ir jų panaudojimas.....	4
2.1.1. SportsML standartas.....	5
2.1.2. XML dokumentą sudarantys blokai.....	8
2.2. XML dokumentų programinės įrangos aksioma.....	9
2.3. Sprendimo būdai.....	10
2.3.1. PĮ platforma.....	10
2.3.2. XML realizavimui pritaikomos kalbos.....	10
2.3.3. XML technologijos duomenų bazių valdymo sistemose.....	11
2.3.4. XML dokumentų ataskaitos.....	15
2.3.5. PĮ įgyvendinimo technologijos.....	15
3. XML dokumentų programinės įrangos architektūra.....	17
3.1. Programų sistemos funkcijos.....	17
3.2. Panaudos atvejų sąrašas.....	19
3.3. Reikalavimai duomenims pagal DB.....	22
3.4. XML dokumentų naudojami standartai.....	23
3.5. Būsenų diagramos.....	23
3.6. Išdėstymo vaizdas.....	24
3.7. XML duomenų vaizdas.....	25
3.8. Aplikacijos paketai.....	27
3.8.1. GUI komponentas.....	28
3.8.2. Dokumentų valdymo komponentas.....	28
3.8.3. DBValdymas komponentas.....	29
3.8.4. Menedžerio komponentas.....	29
3.8.5. Valdymo komponentas.....	30
3.9. Sistemos statinis vaizdas.....	30
3.10. Paketų detalizavimas.....	31
4. Programinės įrangos tyrimo dalis.....	33
4.1. Dokumentų apsikeitimas.....	33
4.2. XML dokumentų analizės įrankis.....	36
4.3. Dokumentų pildymo formos.....	38
5. Programinės įrangos Eksperimentiniai.....	40
5.1. Eksperimentiniai duomenys.....	40
5.2. Eksperimentų sąlygos ir metrikos.....	41
5.2.1. WeatherML eksperimentas.....	42
5.2.2. SportsML eksperimentas.....	43
5.2.3. Individualaus dokumento eksperimentas.....	44
5.3. Sugeneruotos ataskaitos.....	45
6. Rezultatų apibendrinimas ir darbai ateičiai.....	47
7. Išvados.....	50
Literatūra.....	51
Terminų ir santrumpų žodynas.....	53
Priedas A – konferencijos pranešimas.....	54
Priedas B – XML dokumentų analizės rezultatai.....	64

SUMMARY

Nowadays, there is a lot of information in the internet, personal computer, mobile phone and everywhere around us. How we exchange this information? There are many decisions, but the most popular exchange form is using XML document standards. What is more, it will be everything good, but people are, who wants to use them, often faced with certain problems. They are trying get this information from these documents, but they cannot read it or very hard without any special software. Some people do not trust XML documents, because they didn't use any time of them. Where is the problem? Some people could not understand XML document methodology; others did not use special software for special XML documents. It is software for storing and migrating signed XML documents. Special tools can save us time. People can quickly access the documents and information they need. It can help people work together in ways that are most effective for them.

In this mater thesis we will try review the issues of close. We will talk about storing and migrating signed XML documents is developed new software which is very useful for analyzing documents and records. What is more, we will try to create special software which can helps communicate with other people who support this information, share and compare the results between. Other advantage is that this software an open source and it is very useful for non-profit organizations. More other, it can be XML editor and development environment for editing, transforming XML-related technologies. It can help us to do schema designer, file converters (XML to PDF, EXCEL), profilers, supporting, full database integration, support for XPath, Linq and Office Open XML (OOXML) documents.

Experiments comparing developed method with traditional document exchange using XML documents. It showed that software useful and easy work with large document collections and/or large amount of information, generate reports and share information with other users. At the same time software showed that even when new software demonstrates larger functionality and cheapest lowest cost, it defeats other software systems to the usability.

The main application of this work is new software of XML documents analyzers for programming languages using both XPath expressions and XML type annotations (input and output). Such analyzers allow ensure at compile-time valuable properties such as type-safety and optimizations, for safer and more efficient XML processing.

IVADAS

1.1. Darbo tikslai ir rašymo aplinkybės

Šiais technologijų laikais yra plačiai paplitę XML dokumentai, kurių tiesioginė paskirtis yra naudojamas informacijos apsikeitimui. Informacijos kiekis nuolatos auga, todėl sunku ją kontroliuoti. Viskas puiku, kai informacijos daug, tačiau dažnai susiduriame su problema, kai esame nuolatos priversti ieškoti informacijos po didžiulį jos kiekį ir srautą. Su šia problema nuolatos susiduriame tiek darbe, namuose ir dažnai jaučiame nusivylimą, nes bandydami surasti operatyviai informacijos dažnai jos nerandame arba tiesiog sugaištame labai daug laiko. Taigi, dėl šios priežasties nuolatos kenčia mūsų darbo kokybė. Problemos tikriausiai nebūtų, jeigu visa prieinama informacija būtų griežtai standartizuota ir klasifikuota į tipus pagal tam tikras kategorijas, o norint visą tai įgyvendinti, reikia ne tiek daug – taisyklių, palaikymo ir priežiūros. Atsiranda sprendimas, kuris gali pakeisti ir palengvinti reikalingos informacijos paiešką. Paprasčiausiai visą informaciją reikia standartizuoti ir nurodyti kaip ją pateikti kitiems. Visą tai yra paprasta įgyvendinti pasinaudojus XML technologijos pagrindais, kitaip tariant, pritaikyti globalius standartizuotus informacijos apsikeitimo dokumentus.

Autoriui atlikus literatūros analizę ir patyrinėjus šią sritį, buvo pastebėta, kad vartotojai vengia naudotis sukurtais standartizuotais XML dokumentais, kurie yra išties puiki priemonė apsikeitimui svarbia informacija. Viena iš priežasčių gali būti dėl to, kad nėra tinkamai paruoštų ir dokumentuotų XML dokumentų. Kita galima priežastis, kad nėra tam skirtų specifikuotų įrankių. Taigi, atlikus apklausą¹ ir išsamią literatūros analizę² paaiškėjo tos priežastys. Pasirodo yra visiškai kitaip nei manyta. Egzistuoja puikiai dokumentuoti ir aprašyti XML dokumentai, pvz.: SportML (sportiniams įvykiams), WeatherML (orams), FinML (finansams), MarketsML (marketingui), EventsML (žiniasklaidai). Juk jų paskirtis ir yra būtent rinkti ir dalinti informaciją. Vis dėlto, jie yra prastai reklamuojami ir panaudojami, dėl tam tikros priežasties, kad dauguma vartotojų, nepriklausomai nuo jų darbo srities ir patirties, susiduria su tam tikromis problemomis siekiant apdoroti XML dokumentus, o vėliau panaudoti duomenis tolimesniems darbams. Egzistuoja įvairiausių įrankių, kurie sukurti ir pritaikyti pagelbėti, tačiau tos sistemos, įrankiai yra universalūs ir sunkiai integruojami paprastiems vartotojams. Mat joms reikalingas administratorius, įvairūs nustatymai ir techninės įrangos konfigūracijos, o kad būtų galima pakankamai tiksliai ir aiškiai analizuoti turimus rezultatus. Tokia programinė įranga dažniausiai priverčia vartotoją sugaišti daug laiko

¹ Įvykdyta apklausa: <http://www.apklausa.lt/f/standartizuoti-xml-dokumentai-ir-ju-panaudojimas-nk7qkec>

² Literatūros analizė: <http://kristapaitis.byethost24.com/wp-content/plugins/download-monitor/download.php?id=10>

kol įsisavinamas jos funkcijos ir reikalingos operacijos, nuoseklumas, tačiau ir tai dažniausiai nepadedą ir daugelis šių PĮ funkcijų nepanaudojama pagal paskirtį. Kita vertus, nėra jungiamosios grandies tarp vartotojų ir dokumentų valdymo, kuri būtų tinkamai pritaikyta tokiam informacijos srautui valdyti ir komunikuoti tarpusavyje³.

Autorius nusprendė labiau įsigilinti į esamas problemas, pabandyti įgyvendinti savo tikslus ir kitų vartotojų keliamus reikalavimus. Šiame darbe kuriamas pagalbinis XML dokumentų migravimo ir analizės įrankis, gebantis valdyti dokumentų turinį. Panagrinęjus panašių sistemų idėjas – skirtų kurti, apdoroti dokumentus – funkcionalumą ir veikimo principus, buvo nuspręsta sutelkti dėmesį į pačių XML dokumentų informacijos valdymą ir kokybišką jos pateikimą. XML dokumentai pasirinkti kaip forma, kuri yra reikalinga turimai informacijai rinkti ir kaupti.

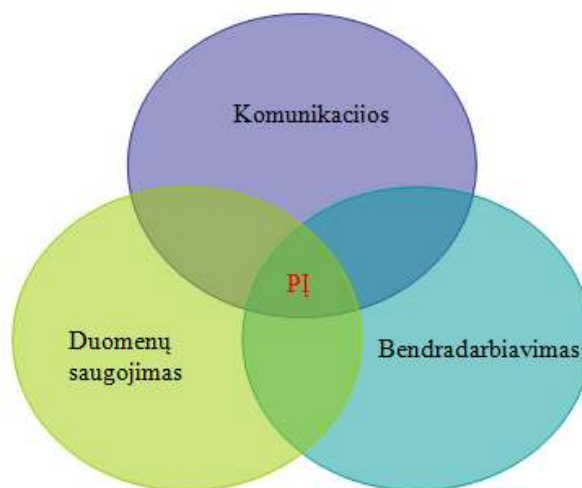
1.2. XML dokumentų valdymo programinė įranga

Įprasti įrankiai, kurie sukurti programuotojams, projektuotojams pvz. Altova, Stylus Studio netinka paprastam vartotojui, nes dažniausiai jie, susiduria su šio tipo dokumentais pirmą kartą, todėl nelabai įsivaizduoja jų funkcionalumo ir panaudijimo. Vienintelis dalykas ko jiems reikia, tai informacija iš paties XML dokumento. Dėl tos priežasties, autorius kaip šios srities ekspertas išvelgia problemą, su kuriomis susiduria paprasti vartotojai nagrinėdami XML dokumentų turinį, nes jiems būdingos sudėtingos (DTD, XSD) statinės schemas (Eric Van der Vlist, 2002). Be to, norint vartotojui analizuoti šiuos dokumentus reikia specialiai tam pritaikytų įrankių, kuriuos aptarsime sekančiame skyriuje. Dažniausiai licenzijos tokių įrankių yra sąlyginai brangios, nes mažai vartotojų specializuojasi XML dokumentų taikymo srityje, o tos programinės įrangos yra didelių „gabaritų“, leidžiančios dirbti programuotojams ir taip keisti XML dokumentų turinį, tačiau viso to paprastam vartotojui nereikia.

Dar vienas iš lemiančių veiksnių šios programinės įrangos kūrimui, tai komunikavimo tarpusavyje problematika. Vartotojams trūksta erdvės, kurioje jie galėtų susijungti į komandas ir tarpusavyje dirbti, bendradarbiauti (Savourel Yves, 2001) atitinkamai kiekvienas pagal tai, kokius duomenys jiems svarbūs, naudingi (sporto federacijos, meteorologai, geodezininkai, analitikai, spauda, žiniasklaida). Deja, šiuo metu nepavyko rasti nė vieno autoriui žinomos ir laisvai prieinamos priemonės įrankio, kuris leistų lengvai XML dokumentus migruoti ir analizuoti.

³ Pavyzdys. Autoriaus dirbamoje įmonėje taip pat egzistuoja duomenų apskaitimo problema tarp skirtingo tipo sistemų. Surenkami įmonėje duomenys yra nuolatos transformuojami į skirtingas formas (XML, SQL, EXCEL, PDF), o tam reikalinga ne viena papildoma programinė įranga, o jas naudojant sugaištamas brangus laikas.

Iliustracijoje (1 pav.) pateikiamas kuriamos PĮ modelio pavyzdys, kuris paliečia tris pačias svarbiausias paprastam vartotojui sritis – bendradarbiavimą, komunikavimą ir duomenų saugojimą. Visos šios trys taikymo sritys yra tarpusavyje labai svarbios, kad pilnai funkcionuotų bet kuri programinė įranga ir tenkintų bet kokio verslo poreikius, dalijantis svarbia informacija (Michael A. Cusumano, 2004).



1 pav. PĮ modelio veikimo principai

1.3. Dokumento turinys

Šį dokumentą kartu su įvadu sudaro šeši skyriai.

Antrame skyriuje yra supažindinama su pagrindine XML dokumentų panaudojimo alternatyva. Skyriuje yra trumpai apibūdinamas analizės įrankis ir nurodomos esminės, visoms panašioms sistemoms būdingos problemos. Apibūdinamas XML dokumentų panaudojimas, veikla, nurodomos jų taikymo sritys, standartai bei tarpusavio skirtumai.

Trečiame skyriuje koncentruojamasi ties sistemos projektavimu. Skyriuje pateikiama metodų klasifikacija, nurodomi proceso žingsniai, detaliam aprašomi jų aspektai ir analizuojama susijusi literatūra.

Ketvirtame skyriuje išsamiai apibūdinama XML dokumentų migravimo problema, apibrėžiamos sąvokos ir pateikiamos pasirašytų XML dokumentų migravimo metodo detalės.

Penktame skyriuje aprašomi atlikti PĮ eksperimentai, kurių metu buvo išanalizuoti XML dokumentai, surinkta svarbi informacija iš įvairių šaltinių ir padalinta kitiems vartotojams. Vėliau jie apdorojami ir pateikiami atitinkama spausdinimo forma.

Paskutiniame skyriuje apibendrinami darbo rezultatai, numatomi darbai ateičiai ir pateikiamos išvados.

2. XML DOKUMENTAI, STANDARTAI IR JŲ PROBLEMOS

Norint išrinkti reikalingus rezultatus iš gausios informacijos, reikia turėti didelį ir patikimą informacijos šaltinį. Informacijos šaltinių šiais laikais yra be galo daug, tačiau jų forma labai įvairi, o norint išrinkti naudingus duomenis nėra paprasta. Natūraliai buvo imtasi spręsti šios problemos ir apsikeitimui pasiūlyta XML technologija, kuri palengvino visą šios informacijos valdymą globaliame tinkle. Deja, pašalinus vieną problemą atsiranda kita, nes XML dokumentai pritaikomi individualiai ir dėl to ko reikia vieniems, gali visiškai netikti kitiems. Dėl tos priežasties reikalingi standartai, kurių atsiradimas pakeičia visiškai kitą šių dokumentų panaudojimą. Padarius apžvalgą galima rasti nemažai sukurtų ir naudojamų XML dokumentams pritaikytų standartų, tačiau atsiranda kita problema analizės įrankiai, kurie sugebėtų valdyti tų dokumentų turinį. Pačių XML dokumentų turinys aprašomas specifiškais schemomis, kurių kontekstas juose atrodo labai painiai. Egzistuoja specifiniai XML dokumentų įrankiai, kurių pagalba galima tuos dokumentus aprašyti ir valdyti, t.y. perkelti duomenis iš vienos formos į kitą. Taigi, nuo čia ir prasideda įdomumas, kaip sukurti įrankį, kurio pagalba būtų galima atlikti išsamią dokumentų analizę ir sėkmingai įgyvendinti informacijos apsikeitimą pasirinktai sričiai arba kitaip tariant, transformuoti XML dokumentus į kitus duomenų formatus (DB, PDF, Excel).

Taigi čia suformuluojamas autoriaus uždavinys ir projektavimo metu iškilusios inžinerijos problemos. Pateikiama galimų sprendimų apžvalga, priimto ir realizuoto sprendimo pagrindimas bei įvertinimas.

2.1. Standartų ideologija ir jų panaudojimas

Dažnai susiduriame su problema, kai esame nuolatos priversti ieškoti informacijos tinkle, tačiau dažnai užtrunkame daug laiko, kol informacija surandama arba išrenkama. Problemos lyg ir nebūtų, jeigu visa prieinama informacija būtų griežtai standartizuota ir suklasifikuota. Norint tai įgyvendinti, reikia griežtų reikalavimų, priežiūros ir palaikymo. Būtent tam yra sukurti XML standartai (Steven Holzner, 2003), tačiau reali situacija yra tokia, kad duomenų surinkimo ir klasifikavimo metodai skiriasi (David Gulbranson et al., 2002) ir duomenys renkami nestandardizuotai, t.y. sukuriami XML dokumentai nestandardizuoti, nors pažvelgus giliau, galima rasti puikių pavyzdžių ir siūlomų standartų naudoti vienai ar kitai pasirinktai sričiai. Įžvelgiama duomenų laikinumo, bei migracijos problema. Šia tema yra organizuojamos internacionalinės konferencijos, taip siekiant surasti ir įgyvendinti visiem tinkamą ir patogų informacijos apsikeitimo būdą (Paulo Carreira, 2004).

Panagrinėsime tam sukurtus XML dokumentų standartus, įvertindami juos pagal vartotojo apibrėžtus kriterijus. Standartų įtraukimas padeda vartotojams greičiau susirasti reikalingus atsakymus į klausimus ir gauti aiškius, palyginamus rezultatus.

Trumpai apžvelkime kuo sukurti standartai pranašesni už kitus ir kur galima juos panaudoti, bei pritaikyti pasirinktai sričiai.

Vieni iš žinomesnių XML dokumentų standartų:

- FinML,
- WeatherML,
- SportsML,
- NewsML,
- EventsML.

2.1.1. SportsML standartas

Kaip pavyzdys pasiimtas autoriaus analizuojamas SportsML (XML dokumentas) standartas. Pirma, tai atviras ir globalus standartas skirtas apsieisti sporto duomenimis (krepšinio, futbolo, olimpiadų, ledo ritulio, golfas, tenisas ir kt.). Šis dokumentas suprojektuotas kaip galima paprasčiau ir suprantamiau. SportsML leidžia keistis varžybų taškais, tvarkaraščiais, statistika, reitingus.

Privalumai:

- Gerai dokumentuoti ir lengva naudoti;
- Atviras standartas;
- Karkasas visų sporto šakų;
- Globalus;
- Apima visus rezultatus (taškai, tvarkaraščiai, reitingai, statistika).

Dokumento praplėtimo galimybės leidžia apimti visas sporto šakas ir kaupti juose duomenis. Be to, palaikomi tokių kaip IPTC standartų NITF (nacionalinis informacijos registras) užtikrina jų tinkamumą, bei viešai pristatomi šių standarto palaikymai. Šie dokumentai naudoja XML technologijas, o tai reiškia, kad apsikeitimas globaliame tinkle labai paprastas ir galima atlikti per RSS kanalus, tereikia tik programinės įrangos apdorojančios dokumentus.

Apžvelkime prieš tai minėto standarto metaduomenis. Kadangi, šis dokumentas remiasi XML standartizuota kalba, tai jo struktūrą atspindi elementai ir jų atributai (W. R. Stanek, 2002).

Kaip pavyzdį panagrinėkime kaip sudarytas SportML, jo duomenų fragmentą. Šiame dokumente panaudojami <sports-event> (lt. sporto įvykis) ir <team> (lt. komanda) raktiniai žodžiai:

```

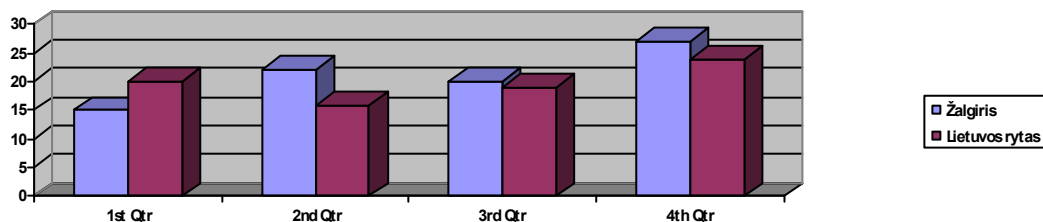
<sports-event>
<sports-metadata
event-status="post-event"
/>
<team>
<team-metadata>
<name
first="Lietuvos rytas"
/>
team-metadata>
<team-stats
score="79"
event-outcome="loss"
/>
team>
<team>
<team-metadata>
<name
first="Žalgiris"
/>
team-metadata>
<team-stats
score="84"
event-outcome="win"
/>
team>
sports-event>

```

Tačiau galima visą šį dokumento turinį pateikti vartotojui aiškesniame pavidale panaudojus įrankius, kurie geba skaityti logines struktūras ir atvaizduoti rezultatus lentelėje² arba grafiniame pavidale:

1 lentelė. Suminiai rezultatai

Komanda	Taškai
Žalgiris	84
Lietuvos rytas	79



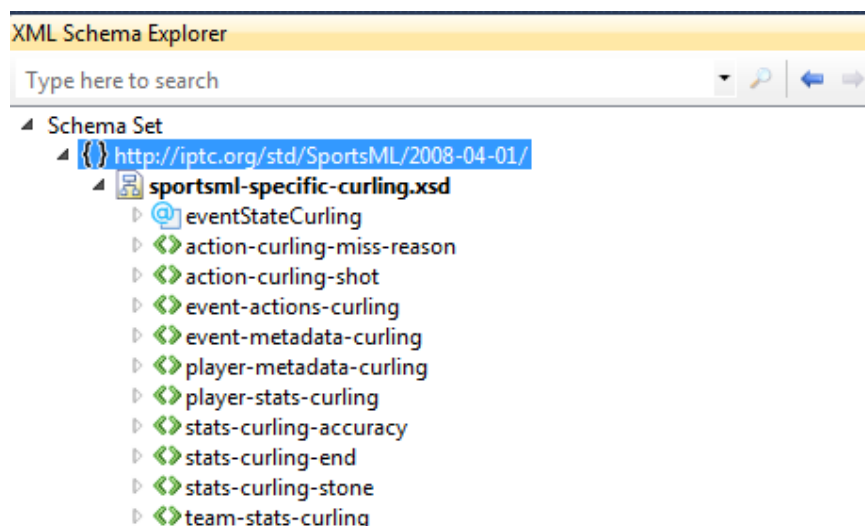
2 pav. Pelnytų taškų suvestinė pagal ketvirčius

Pagrindinė SportsML struktūra susideda iš šių objektų:

- <sports-event>
- <tournament>
- <schedule>
- <standing>
- <statistic>
- <article>

Pirmieji penki objektai yra pagrindiniai ir tarpusavyje kombinuojami su kitais susijusiais elementais tokiais kaip <player> (lt. žaidėjas), <team> (lt. komanda). SportsML yra gerai struktūrizuotas dokumentas, pagal kurį galima sudėti įvairių varžybų rezultatus. Be to, struktūra ir duomenys palaikomi ir patvirtinti įvairių partnerių, bei tarpininkų. Tokiuose dokumentuose galima talpinti ne tik susijusia informaciją su pačiomis rungtynėmis ar varžybomis, tačiau galima juos individualiai praplėsti pagal savo poreikius ir detalizuoti (lygas, asociacijas, konferencijas). Tokiu būdu renkant visą svarbia informaciją, galima nustatyti kiekvieno žaidėjo naudingumą, palyginti jų rezultatyvumą ir kitus duomenis.

Vienas iš svarbių ir nemažai laiko atimančių architektūros sprendimų yra dokumento tipo nustatymas, todėl čia yra panaudojamas XML dokumento žodynas (DTD), kuriame tiksliai nusakomas to dokumento kiekvieno elemento tipas.



3 pav. Suskleista XSD schema

Tokiose schemose yra aprašomi visi galimi įvykių atvejai. Pvz.: komandos, žaidėjai, reitingai, statistika, rezultatai, laikai, vietos. Žodžiu, tokie dokumentai yra pilnai aprašyti su įvairiais galimais rezultatų pateikimo atvejais.

XML failo su įterptu DTD aprašu pavyzdys (SportsML):

```
<!ENTITY % event-state-attributes-curling "  
  stone-idref          IDREF          #IMPLIED  
  player-idref         IDREF          #REQUIRED  
  team-idref           IDREF          #IMPLIED  
  end-number           CDATA          #IMPLIED  
">  
  
<!ELEMENT event-metadata-curling EMPTY>  
<!ATTLIST event-metadata-curling  
  %global-attributes;  
  sheet      CDATA      #IMPLIED  
  draw       CDATA      #IMPLIED  
>  
  
<!ELEMENT team-stats-curling (stats-curling-stone*, stats-curling-end*, stats-curling-accuracy*)>  
<!ATTLIST team-stats-curling  
  %global-attributes;  
  %coverage-attributes;  
>  
  
<!ELEMENT stats-curling-stone EMPTY>  
<!ATTLIST stats-curling-stone  
  %global-attributes;  
  %coverage-attributes;  
  stone-number      (1|2|3|4|5|6|7|8) #REQUIRED  
  stone-color       CDATA          #REQUIRED  
>  
  
<!ELEMENT stats-curling-end EMPTY>  
<!ATTLIST stats-curling-end  
  %global-attributes;  
  %coverage-attributes;  
  end-number        CDATA          #IMPLIED  
  score             (0|1|2|3|4|5|6|7|8) #IMPLIED  
  throw-order       (first|last)   #IMPLIED  
  time-used         CDATA          #IMPLIED  
>
```

Viršuje pateikta vienas iš pavyzdžių SportsML (DTD) ir tai tik aprašymo fragmento dalis. Galime susidaryti vaizdą, kad tokie XML dokumentai leidžia pilnai aprašyti visus galimus ir įmanomus duomenų tipus ir jų informaciją (W. R. Stanek, 2002). Žinant duomenų tipus galima nesunkiai pritaikyti informacijos rinkimo ir analizės algoritmus pasirinktai sričiai.

2.1.2. XML dokumentą sudarantys blokai

DTD požiūriu XML dokumentą sudaro penki pagrindiniai blokai:

- a) Elementai
- b) Atributai
- c) Esybės
- d) PCDATA
- e) CDATA

Elementai yra svarbiausi XML dokumento komponentai (paskaitoje apie HTML buvo vadinami žymėmis). Elementai gali apimti tekstą, kitus elementus, o gali būti tušti. Elementų pavyzdžiai:

```
<body>tekstas</body>  
<message>tekstas</message>
```

Atributuose pateikiama papildoma informacija apie elementus. Atributai visada talpinami atidarancioje elemento žymėje. Atributai visuomet sudaryti iš porų vardas/reikšmė.

Pvz.:

```
.
```

Kai kurie simboliai XML kalboje turi specialią reikšmę, pvz., simbolis "<" reiškia XML žymės pradžią. Jei reikia tekste įterpti šį simbolį, vietoje jo rašoma tam tikra simbolių kombinacija, vadinama esybe. Apdorojant XML dokumentą XML analizatoriaus programa, esybės pakeičiamos atitinkamais simboliais.

PCDATA reiškia analizuojami simboliniai duomenys (angl. parsed character data). Juose sutiktos elementų žymės traktuojamos kaip elementai, o juose esančios esybės pakeičiamos atitinkamais simboliais. Taigi duomenyse negali būti simbolių "<", ">", "&", jie turi būti pakeisti atitinkamomis esybėmis.

CDATA reiškia tiesiog simbolinius duomenis. Juose esančios elementų žymės nėra traktuojamos kaip elementai, o esybės nėra pakeičiamos.

2.2. XML dokumentų programinės įrangos aksioma

Išanalizavus esamus standartus, jų privalumus ir trūkumus, kyla natūralus poreikis programinės įrangos, kuri analizuotų tuos dokumentus, nes paprastam vartotojui jų struktūra yra sunkiai perskaitoma, o tuo labiau panaudojami jų duomenys išsamesnei analizei. Šiuolaikinių technologijų amžiuje yra apstu informacijos apie tų dokumentų surinkimą. Programuotojai gali laisvai rinktis kas jiems geriau ir lengviau įgyvendinama. Autoriaus prioritetai paskirstyti atlikus išsamią analizę buvo .NET Framework 3.0, 3.5 platformai, C#, Linq kalboms. Duomenų saugykla sukurta ant Microsoft SQL Server platformos. Kituose skyriuose papasakosime plačiau apie pasirinktų technologijų įgyvendinimą ir jų pasirinkimo kriterijus. Programinės įrangos vienas iš tikslų ir prioritetų sukurti atviro kodo programinę įrangą, bei leisti ja dalintis, kad ji galėtų funkcionuoti ir būti tobulinama be autoriaus žinios ir pastangų, taip kaip vartotojams patogiau. Architektūroje naudojamos tik nemokamos ir laisvai taikomos komerciniams tikslams bibliotekos.

2.3. Sprendimo būdai

Pirma, kuriant bet kokią programinę įrangą galima pasirinkti iš labai didelės gausos šiuolaikinių technologijų. Sprendžiant realizavimo klausimą su XML dokumentų apdorojimais ir saugyklomis, susiduriama su tam tikrais sunkumais. Kaip pasirinkti tinkamą technologiją, kad tai būtų lengviau realizuoti ir galutinis rezultatas būtų patogesnis ir priimtinesnis galutiniam vartotojui. Autoriui dirbant ir nemažai patirties turint IT srityje buvo perskaityta nemažai literatūros (Vladimir Geroimenko, 2004). Vladimir Geroimenko jau seniai dirba ir domisi XML dokumentų kilme, jų panaudojimu, schemomis, jų apdirbimo technologijomis. Pagrindinė mintis, kad technologijos sparčiai vystosi ir auga, todėl spręsti kuri geriausia technologija yra labai sudėtinga. Pasak knygos autoriaus (Vladimir Gerosimenko, 2004) verčiau rinktis tokį principą, kuris yra šiuo metu populiariausias ir naujausias, tik tokiu atveju galima tikėtis pranašumo prieš kitas technologijas ir rasti daugiau pagalbos sprendžiant vieną ar kitą problemą.

2.3.1. PĮ platforma

Pasaulyje labiausiai paplitusios yra Microsoft operacinės sistemos, todėl buvo nuspręsta remiantis dideliu vartotojų skaičiumi, kurti PĮ pasinaudojant .NET Framework 3.0 platforma. Microsoft.NET Framework yra Microsoft Windows operacinės sistemos komponentas. Jis suteikia kitoms programoms galimybę naudotis daugybe jau paruoštu įvairių bibliotekų (pvz. duomenų bazių komponentus, formų komponentus). Be to, šis komponentas ir tvarko programos kodą jos vykdymo metu, jei programa parašyta specialiai šiam paketui. Tai reiškia, kad programa vienodai gerai turėtų veikti įvairiose platformose; nėra būtinybės 64-ių bitų procesoriams skirtą CIL programą perkompiliuoti į 32-ų bitų skirtą procesoriams programą. Visa tai atliekama labai greitai ir automatiškai.

2.3.2. XML realizavimui pritaikomos kalbos

Apžvelgiant literatūrą, sprendžiant vieną ar kitą klausimą, susiduriant su XML dokumentų duomenų išrinkimu ir apdorojimu buvo pastebėta, kad idealiausiai tinkantis sprendimas yra .NET Framework 3.0 naujovė - LINQ (Language Integrated Query) technologija, leidžianti operuoti duomenimis naudojant deklaratyvią sintaksę panašią į SQL. Didelis LINQ technologijos privalumas apart deklaratyvumo yra gebėjimas dirbti su skirtingais duomenų šaltiniais: objektais, duomenų bazėmis, XML, DataSet. Kadangi numatome PĮ toliau tobulinti ir plėsti, todėl universali XML duomenų išrinkimo kalba pagerins integraciją tarp skirtingų reliacinių duomenų bazių (Oracle, MySQL, Prosgate).

Taip pat, pasirinkta XPath, tai kelio standartas duomenų struktūrai XML dokumente pasiekti. XPath sintaksė kiek primena failų sistemose naudojamą kelią dokumentams aplanke rasti, tačiau yra daug lankstesnė ir numato atvejus kuomet tame pačiame aplanke (pavyzdžiui, narių sąrašė) yra vienodus vardus turintys elementai. Galimas elementų indeksavimas bei paieška pagal atributų reikšmes. XPATH gali būti naudojama skyrium arba kaip sudėtingesnės XSLT transformacijų kalbos dalis. XPath ašis apibrėžia aibę mazgų, susijusių su konteksto mazgu. Kiekvienos ašies mazgų aibė yra vieno tipo: elementų, atributų arba vardų srities.

XPath egzistuoja tokios ašys:

- *child* – aibė elementų, kurių tėvas yra konteksto mazgas.
- *descendant* – aibė elementų, kurių tėvas, tėvo tėvas ir t. t. yra konteksto mazgas.
- *parent* – elementas, kuris yra konteksto mazgo tėvas.
- *ancestor* – aibė elementų: konteksto mazgo tėvas, tėvo tėvas ir t. t.
- *following - sibling* – aibė po konteksto mazgo einančių elementų, kurių tėvas tas pats kaip ir konteksto mazgo.
- *preceding - sibling* – aibė prieš konteksto mazgą einančių elementų, kurių tėvas tas pats kaip ir konteksto mazgo.
- *following* – aibė elementų, esančių po konteksto mazgo.
- *preceding* – aibė elementų, esančių prieš konteksto mazgą.
- *attribute* – aibė konteksto mazgo atributų.
- *namespace* – aibė vardų sričių mazgų, susietų su konteksto mazgu.
- *self* – konteksto mazgas.
- *descendent - or - self* – *self* ir *descendent* elementų sąjunga.
- *ancestor - or - self* – *self* ir *ancestor* elementų sąjunga.

Aplikacija kuriama C# kalba, tai objektinė orientuota programavimo kalba, sukurta Microsoft kompanijoje kaip dalis .NET iniciatyvos. Kalba paremta C++ bei Java kalbomis. Be to, kalba kurta balansuojant tarp galingumo (C++ įtaka) bei greito programavimo (Java, Visual Basic įtaka), beje puikiai priderinta kurti framework platformai ir turi gerų bibliotekų dirbti su XML dokumentais.

2.3.3. XML technologijos duomenų bazių valdymo sistemose

Norint atlikti skirtingų tipų duomenų šaltinių integraciją, visų pirma reikia turėti visiems integruojamiems šaltiniams bendrą schemą, kuri leistų atlikti integraciją. Šiame darbe

nusprendėme pasirinkti XML kalbą ir šios kalbos struktūrą apibrėžiančias XML schemas. Panagrinėsime XML kalbos ir schemų privalumus bei trūkumus, aptarsime jų sandarą bei panaudojimą.

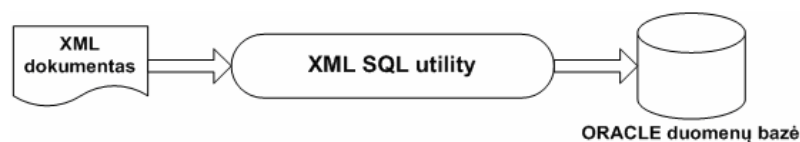
Oracle duomenų bazė ir XML panaudojimo galimybės

Oracle turi XML failų nagrinėjimo priemones (angl. *parsers*) Java, C, C++ ir PL/SQL kalboms, todėl programuojant šiomis kalbomis XML panaudojimas yra lengvesnis. Naudojant XML SQL Utility galima vykdyti SQL užklausas ir gauti rezultatus XML dokumentą iš suformuotų užklausų rezultatų.



4 pav. XML dokumento formavimas naudojant XML SQL Utility

Gauto XML dokumento struktūra atitinka Oracle DB struktūrą, kuri buvo nurodyta formuojant SQL užklausa. Duomenų bazės stulpelių pavadinimai atitinka XML elementų grupių pavadinimus, o reikšmės yra priskiriamos vidiniams elementams. Taip pat naudojantis XML SQL Utility galima ir surašyti XML duomenis į Oracle duomenų bazę.



5 pav. XML dokumento duomenų surašymas į Oracle DB

MS SQL Server duomenų bazė ir XML panaudojimo galimybės

MS SQL Server duomenų bazės duomenis į XML galima perduoti trimis būdais:

- Baigiant SELECT sakinį FOR XML;
- XPath užklausų priemonių pagalba, kurios komentuoja XML schemas;
- OpenXML funkcija procedūrose.

OpenXML funkcija dažniausiai naudojama atvaizduoti XML dokumento duomenims į sąmyšinės bazės lenteles.

MS SQL Server yra galimos tokios XML savybės:

- HTTP pasiekiamumas (rašant SQL sakinį);
- Formuoti XML galima SELECT sakinio pagalba (FOR XML);
- XML views (nurodoma kaip sąryšiniai duomenys turi būti atvaizdinti);
- XPath queries (nurodoma kaip duomenys turi būti paimti iš DB);
- OpenXML (nurodo, kaip duomenys turi būti atvaizduoti XML ir
- ML failo duomenis atvaizduoti duomenų bazėje lentelėse);
- OLE DB ir ADO pasiekiamumas;
- ML iš DB ir surašant XML duomenis į DB).

Oracle ir MS SQL Server teikiamų savybių palyginimas

2 lentelė. MS SQL server ir Oracle palyginimas

Savybės	MS SQL Server 2005	Oracle
<p>Deklaratyvus žymėjimas laisvai susietų verslo sistemų (ši savybė suteikia dvikryptį žymėjimą tarp XML ir sąrašinių duomenų vaizdų).</p> <p>SQL Server 2005 turi integruotas XML savybes, kurios leidžia priėjimą prie sistemos su minimaliu programavimu.</p> <p>Oracle neturi integruotų XML specifinių savybių.</p>	<p>XML Views XPath support Transact-SQL FOR XML išplėtimai Transact-SQL OPENXML išplėtimai</p>	<p>Neturi specifinių savybių. Reikia rašyti papildomas programas.</p>
<p>Vietiniai XML išplėtimai skirti SQL ir leidžiantys vartotojui vykdyti aplikaciją greitai ir be papildomo programavimo.</p> <p>SQL Server 2005 teikia išplėtimus SQL kalbai, kurie gali būti panaudoti gražinti XML duomenis iš standartinių SQL užklausų.</p> <p>Oracle teikia serverio pagalbinius įrankius, kuriuos vykdant neišplečiama SQL ir reikalaujama papildomo programavimo gražinant XML duomenis.</p>	<p>Transact-SQL FOR XML išplėtimai Transact-SQL OPENXML išplėtimai</p>	<p>Neturi specifinių savybių. Reikia rašyti papildomas programas naudojant XSQL Utility.</p>
<p>Įvairūs XML prieigos metodai, kurie palaiko lankstumą, leidžiantį programuotojams pasiekti XML duomenis per tinklo prisijungimus.</p>	<p>HTTP prieiga OLE DB/ADO prieiga JDBC</p>	<p>HTTP prieiga OLE DB/ADO prieiga JDBC</p>

SQL Server 2005 ir Oracle palaiko įvairius metodus leidžiančius pasiekti XML duomenis, esančius duomenų bazėse.		
XML šablonai SQL Server ir Oracle palaiko išsaugotus XML užklausų šablonus serveryje su laisvai pasirenkamais parametrais.	URL/HTTP prieiga XML šablonai	URL/HTTP prieiga XSQL šablonai
Saugi Web aplikacijų prieiga prie XML duomenų SQL Server 2005 palaiko saugumą, kuris gali būti valdomas individualios lentelės lygyje. Oracle palaiko apribotą saugumą tik duomenų bazės vartotojo, bet ne individualios lentelės lygio.	Duomenų bazės vartotojo lygmens saugumas Duomenų bazės objekto lygmens saugumas	Duomenų bazės vartotojo lygmens saugumas
Galimybės detalesnei paieškai sudėtingiems XML dokumentams. SQL Server 2005 ir Oracle teikia duomenų bazės palaikomą sudėtingų tekstinių dokumentų paiešką apimančią ir XML.	Pilna teksto paieška	Tarpinė teksto paieška su segment savybėmis.
XML failų nagrinėjimo priemonės (angl. <i>parsers</i>) (skirti programuotojams, kurie siekia didesnės galios ir lankstumo, kai kuriamos XML aplikacijos).	MSXML COM priėjimo parserių palaiko daugelis kalbų	Oracle parserius palaiko daugelis kalbų

Atlikus palyginamąją analizę paaiškėjo, kad geresnes savybes ir suderinamumą turi MS SQL serveris. Microsoft SQL Server 2005 siūlo XML geriausias savybes skirtas tiek Web, aplikacijoms ir duomenų bazių programuotojams. Web ir aplikacijų programuotojams nereikia mokintis duomenų bazių programavimo subtilybių norint pasinaudoti SQL Server 2005 XML privalumais, nes SQL Server teikia standartines XML konstrukcijas ir palaiko XML duomenų tipą, ko neturi MySQL ir Oracle. Duomenų bazių programuotojams nereikia išmanyti XML programavimo subtilybių, nes jie gali naudotis FOR XML ar OpenXML siekdami manipuluoti XML duomenimis.

Oracle labiau skirtas naudoti patyrusiems programuotojams. Norint pasiekti sprendžiamų uždavinių rezultatą tenka naudotis keliomis priemonėmis arba kurti specifinius įrankius.

2.3.4. XML dokumentų ataskaitos

CrystalReports ataskaitos yra tam tikro formato dokumentas, kuriame atvaizduotas visos DB ar jos atskirų dalių turinys. Pagrindinė ataskaitos paskirtis - pateikti duomenis (rezultatus) patogioje vartotojui formoje, t.y. taip, kad juos būtų galima lengvai analizuoti, apibendrinti ir padaryti teisingas išvadas. CrystalReports įgalina gana paprastai sukurti ataskaitą. Duomenis leidžiama įvairiai rūšiuoti, grupuoti, atlikti su jais įvairius skaičiavimus. Paruošta ataskaita gali būti peržiūrima kompiuterio ekrane, išsaugoma išorinėje atmintyje kaip atskiras failas, spausdinama popieriuje, po to įvairiais būdais dauginama. Ataskaitą galima sukurti dviem būdais:

1. Naudojantis standartinėmis formomis;
2. Pačiam vartotojui pasirenkant ataskaitos formą.

Pirmasis būdas gerokai apriboja vartotojo pasirinkimą, kadangi ataskaita kuriama pagal sistemos siūlomą formatą. Žymiai daugiau galimybių atsiranda formuojant ataskaitą antruoju būdu. Čia vartotojas gali laisvai pasirinkti jam priimtinausią ataskaitos formatą (PDF, Excel, XML). Be to, į pagalbą ateina tam tikslui skirti specialūs instrumentai – ataskaitų dizaineriai. Modernios DBVS taip pat leidžia duomenis pateikti įvairiausių formatų diagramų pavidalu.

Kiekviena ataskaita turi tam tikrą struktūrą, į kurią privalu atsižvelgti. Ataskaitoje išskiriamos tokios trys zonos: antraštės zona, pagrindinė zona, pabaigos zona. Antraštės zonoje rašoma informacija, kuri turi būti viso dokumento pradžioje. Jei dokumentą sudaro keli puslapiai, tai nurodoma ir informacija, kuri turi būti kiekvieno puslapio pradžioje. Pagrindinė zona apima eilutes, kurios dažniausiai vaizduoja vieną duomenų bazės įrašą. Ataskaitos pabaigos zona analogiška antraštės zonai, tik jos turinys vaizduojamas dokumento (ar puslapio) pabaigoje.

2.3.5. PĮ įgyvendinimo technologijos

Autorius realizuodamas PĮ pasirinko šiuolaikines technologijas, kurios šiuo metu yra vienos iš populiariausių sprendžiant panašius sprendimus. Be to, naujausios technologijos yra pranašesnės ir perspektyvesnės žvelgiant į ateitį, nes kuriamas projektas yra atvirojo kodo, o tai neatsiejama nuo tobulinamo darbo ir tolimesnio PĮ palaikymo. Pagrindinė platforma pasirinkta .NET Framework aplinkoje dėl jos funkcionalumo ir universalumo, bei daugybės

sukurtų bibliotekų padedančių apdoroti XML duomenis. Žvelgiant ir į ateities planus, nurodomos kitos pasirinktos reliacinės bazės (MySQL, Oracle). Operacinė sistema naudojama, bet kuri iš Microsoft šeimos (Windows XP, Vista, 7).

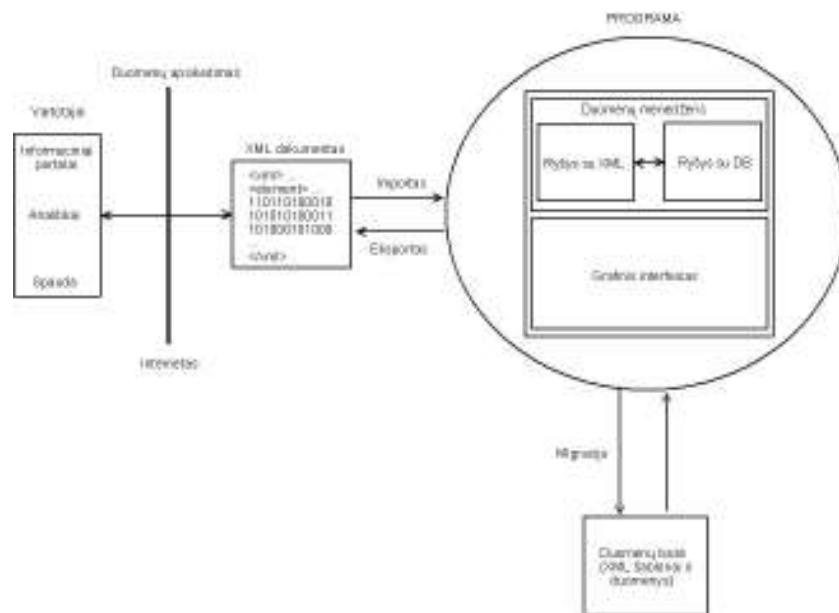
<i>Programavimo įrankis</i>	
<i>Programavimas</i>	
<i>Kitos technologijos, kalbos</i>	CrystalReports, Linq, Xpath, C#, XML
<i>Reliacinės duomenų bazės</i>	  
<i>Operacinė sistema</i>	

6 pav. Technologiniai sprendimai

3. XML DOKUMENTŲ PROGRAMNĖS ĮRANGOS ARCHITEKTŪRA

Šiame skyriuje pateikiama projektinės dalies – programinės įrangos techninė dokumentacija. Pateikiamos PĮ konteksto ir panaudos atvejų diagramos, kurios atspindi pagrindinius įrankio veikimo principus ir funkcionalumą.

Šios PĮ privalumas yra universalumas ir lankstumas. Galimybė apsiraišyti savo norimą XML standartą, jį tvarkyti. Pagrindinis programinės įrangos modulis leis valdyti duomenų bazę realiu laiku ir kontroliuoti duomenų mainus tarpusavyje su kitais vartotojais. Programinės įrangos kontekstinis modelis pateiktas (7) paveikslėlyje.



7 pav. Programinės įrangos panaudojimo atvejai

3.1. Programų sistemos funkcijos

Sistemos vartotojai naudodamiesi programinę įrangą gali naudotis teikiamomis jos funkcijomis. Pagrindinės sistemai funkcijos:

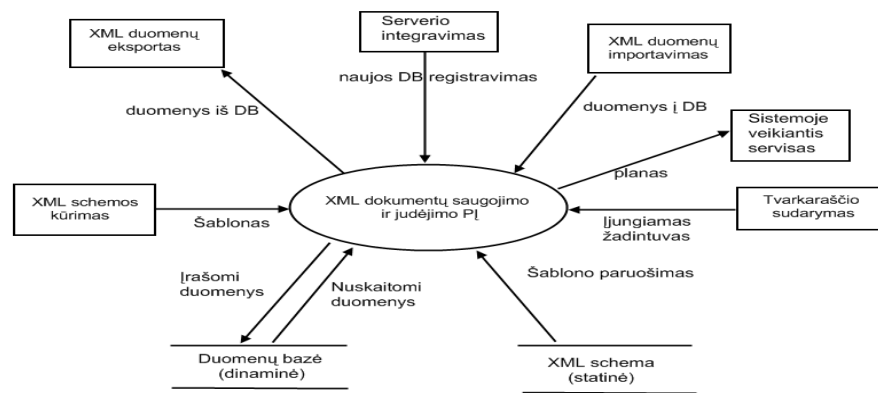
- **Įvairūs veiksmai su sistemos vartotojais.**
Sistemos administratorius galės aprašyti naujus vartotojus, juos redaguoti, šalinti.
- **Duomenų importas.**
Sistemos vartotojai galės atnaujinti duomenų bazės turinį pasinaudojus XML standartų dokumentus.
- **Duomenų eksportas.**
Sistemos vartotojai galės eksportuoti duomenis į XML dokumentus iš duomenų bazės arba PĮ pati kas kažkiek laiko atnaujinti nurodytą XML dokumentą duomenimis.

- **Galimybė pasirašyti naują XML dokumentą.**

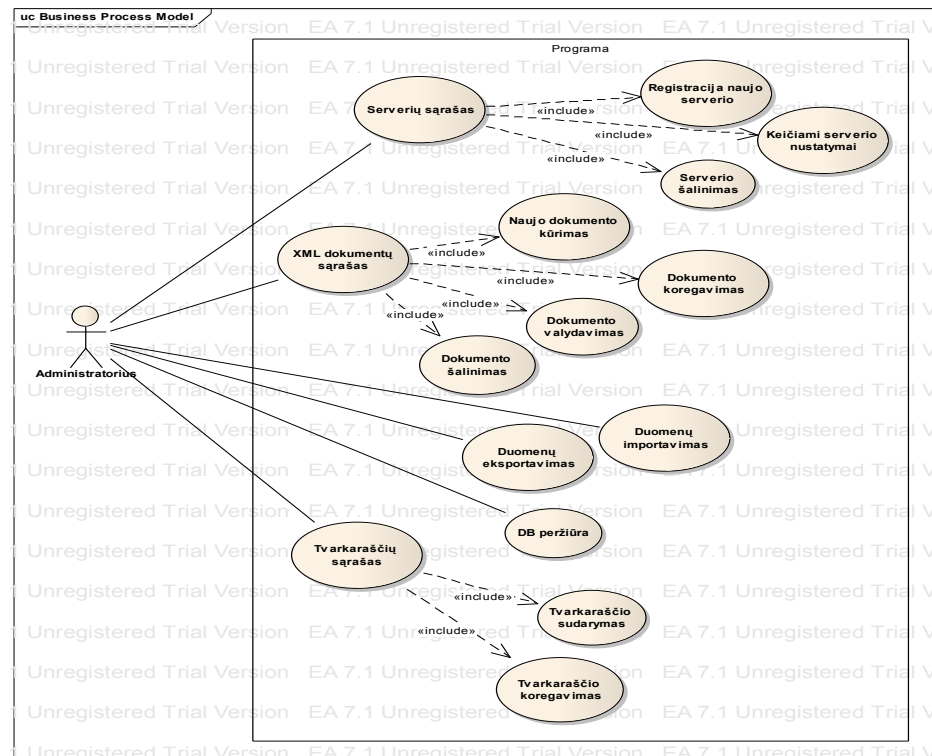
Sistemos vartotojas galės susikurti naujus dokumentus pagal egzistuojančius standartus, jas redaguoti, šalinti, atlikti paiešką pagal norimus duomenis.

Papildomos sistemos funkcijos:

- Kurti, redaguoti, šalinti naujas varžybų komandas.
- Įvesti naujus žaidėjus, teisėjus, trenerius, bei juos redaguoti.
- Rinkti statistinę informaciją varžybų metu.
- Kurti analizės lenteles, bei ataskaitas.
- Transliuoti varžybų rezultatus tinkle.



8 pav. Konteksto diagrama



9 pav. Panaudos atvejų diagrama

3.2. Panaudos atvejų sąrašas:

3 lentelė.

1. PANAUDOJIMO ATVEJIS: Serverių sąrašas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Prisijungęs prie sistemos vartotojas pasirenka kokį serverį naudos dirbant su duomenimis.
Prieš sąlyga:	Vartotojas prisijungia prie programos.
Sužadinimo sąlyga:	Kuriamas naujas XML dokumentas.
Po-sąlyga:	Programos lange parodomas MS SQL serverių sąrašas.

4 lentelė.

2. PANAUDOJIMO ATVEJIS: Naujo DB serverio registracija	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Jeigu yra naudojami keli DB serveriai arba jungiamasi prie serverio per tinklą, galima nurodyti programai kelis serverio adresus arba tiesiog įtraukti naują serverį.
Prieš sąlyga:	PĮ neturi jokio ryšio su DB serveriu (MS SQL).
Sužadinimo sąlyga:	Reikia naujo DB serverio, administratorius suveda naujo serverio prisijungimo duomenis.
Po-sąlyga:	Sistemoje užregistruojamas naujas serveris. Nustatymai išsaugomi XML dokumente.

5 lentelė.

3. PANAUDOJIMO ATVEJIS: Keičiami serverio nustatymai	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Vartotojas prisijungęs prie PĮ, gauna informaciją apie MS SQL nepasiekiamumą. Keičiami MS SQL serverio prisijungimo duomenys arba pereina į antra PA.
Prieš sąlyga:	PĮ nepasiekia MS SQL serverio.
Sužadinimo sąlyga:	Atsijungęs serveris, keičiami nustatymai.
Po-sąlyga:	PĮ prisiregistruoja prie MS SQL DB serverio.

6 lentelė.

4. PANAUDOJIMO ATVEJIS: Šalinamas DB serveris iš sąrašo	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Panaikinamas serveris iš programinės įrangos DB sąrašo.
Prieš sąlyga:	Senas DB (MS SQL) serveris, kurio nustatymai visiškai nebereikalingi.
Sužadinimo sąlyga:	Pasirenkamas serveris iš esančio sąrašo.
Po-sąlyga:	Ištrinami serverio nustatymai ir pats serveris iš prisijungimo sąrašo.

7 lentelė.

5. PANAUDOJIMO ATVEJIS: XML dokumentų sąrašas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Vartotojas dirbdamas su duomenimis turi nurodyti XML dokumentą pagal kurį šabloną bus dirbama.

Prieš sąlyga:	Darbo pradžia su duomenimis.
Sužadinimo sąlyga:	Pasirenkamas XML dokumentas, kuris bus naudojamas darbe.
Po-sąlyga:	Pateikiami visi administratoriaus sukurti arba naudojami XML dokumentai.

8 lentelė.

6. PANAUDOJIMO ATVEJIS: Naujo XML dokumento kūrimas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Administratorius sukuria naują XML, kad galėtų esant reikalui perkelti duomenis iš XML dokumento į DB serverį arba atvirkščiai.
Prieš sąlyga:	Pasirenkamas XML dokumentų sąrašas.
Sužadinimo sąlyga:	Suvedamas dokumento pavadinimas, užrašoma jo struktūra (šaknis, elementai, atributai).
Po-sąlyga:	Sistema priregistruoja naują XML dokumentą, o DB serveryje sukuriamą lentelė duomenų išsaugojimui. Išsaugoma XML dokumento struktūra.

9 lentelė.

7. PANAUDOJIMO ATVEJIS: XML dokumento tikrinimas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Prieš keliant duomenis arba importavus naujo XML dokumentą į sistemą, rekomenduojama dokumentą patikrinti ir peržiūrėti kaip jo struktūra atitinka su DB esančiais duomenimis.
Prieš sąlyga:	Sistemoje egzistuoja bent vienas XML dokumentas.
Sužadinimo sąlyga:	Atsirado naujas XML dokumentas arba nevisi duomenys persikelia į DB.
Po-sąlyga:	Pateikiamas procentinis struktūros sutapimas su DB, bei koregavimo variantai.

10 lentelė.

8. PANAUDOJIMO ATVEJIS: XML dokumento koregavimas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Pasikeitus dokumento struktūrai, atitinkamai reikia suderinti jį su DB esančiais duomenimis.
Prieš sąlyga:	Egzistuoja XML dokumentas ir jo struktūra sukurta DB serveryje.
Sužadinimo sąlyga:	Suvedami nauji atributai, elementai.
Po-sąlyga:	Suformuojama nauja XML dokumento schema – struktūra.

11 lentelė.

9. PANAUDOJIMO ATVEJIS: XML dokumento šalinimas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Administratorius reguliariai patikrina DB ir atrenka nereikalingus dokumentus.
Prieš sąlyga:	Egzistuoja XML dokumentas ir duomenys duomenų bazėje.
Sužadinimo sąlyga:	Pasirenkamas XML dokumentų sąrašas. Tikrinami DB esantys įrašai.

Po-sąlyga:	Ištrinami duomenys esantys DB serveryje susiję su šalinamu XML dokumentu. Atnaujinamas dokumentų sąrašas.
-------------------	---

12 lentelė.

10. PANAUDOJIMO ATVEJIS: XML duomenų importas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Administratorius gauna naują XML dokumentą su naujais duomenimis.
Prieš sąlyga:	XML dokumento struktūra egzistuoja DB serveryje.
Sužadinimo sąlyga:	Pasirenkamas dokumentas, kurio duomenys bus importuojami į DB.
Po-sąlyga:	Duomenys įrašomi į DB.

13 lentelė.

11. PANAUDOJIMO ATVEJIS: XML duomenų eksportas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Apima procesą, kurio metu administratorius ištraukia duomenis iš DB serverio į XML dokumentą.
Prieš sąlyga:	Egzistuoja XML dokumento struktūra, bei duomenys DB.
Sužadinimo sąlyga:	Pasirenkamas šablonas pagal kurį PĮ eksportuos duomenis esančius MS SQL serveryje.
Po-sąlyga:	Sukuriamas naujas XML dokumentas.

14 lentelė.

12. PANAUDOJIMO ATVEJIS: Tvarkaraščio sąrašas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Automatizuotų užduočių sąrašas, kurios yra sukurtos.
Prieš sąlyga:	Užregistruota bent viena užduotis.
Sužadinimo sąlyga:	Pasirenkamas užduočių tvarkaraštis.
Po-sąlyga:	Atspausdinamos visos sistemoje esančios užduotys, bei jų būsenos (aktyvios, išjungtos).

15 lentelė.

13. PANAUDOJIMO ATVEJIS: Tvarkaraščio sudarymas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Automatizuotas duomenų importas arba eksportas. Administratorius nustato įvykdymo laiką ir programa savarankiškai atlieka nustatytas komandas, veiksmus.
Prieš sąlyga:	Egzistuoja bent vienas XML dokumentas.
Sužadinimo sąlyga:	Įvedamas laikas, data ir pasikartojimo periodas, kuriuo metu reikia įvykdyti užduoti (importas, eksportas), pasirenkamas XML dokumentas.
Po-sąlyga:	Įjungiamas servisas.

16 lentelė.

14. PANAUDOJIMO ATVEJIS: Tvarkaraščio koregavimas	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Apima procesą, kurio metu laivo vado pavaduotojas registruoja informaciją apie laivo personalą.

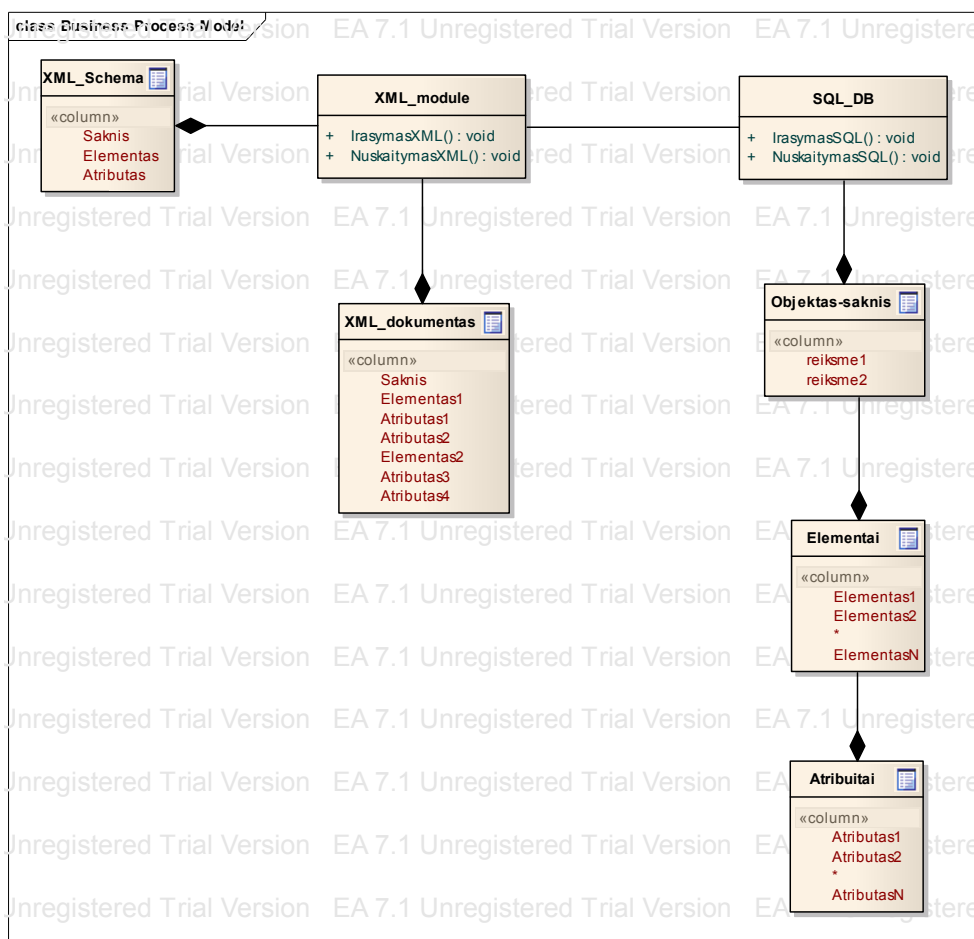
Prieš sąlyga:	Sistemoje yra nors vienas sudarytas tvarkaraštis.
Sužadinimo sąlyga:	Pasikeitė įvykdymo laikas arba pats dokumentas.
Po-sąlyga:	Atnaujinama duomenų bazė. Servisas gauna naujus įvykdymo duomenis.

17 lentelė.

15. PANAUDOJIMO ATVEJIS: DB peržiūra	
Vartotojas/Aktorius:	Administratorius
Aprašas:	Administratorius peržiūri XML dokumento įrašytus duomenis MS SQL serveryje.
Prieš sąlyga:	Pasirenkamas XML dokumentas ir DB serveris.
Sužadinimo sąlyga:	Atsirado naujas asmuo, kuris bus personalo narys (dalyvaus apskaitoje).
Po-sąlyga:	Parodomi visi duomenų bazės įrašai, kurie buvo įrašyti iš pasirinkto dokumento.

3.3. Reikalavimai duomenims pagal DB

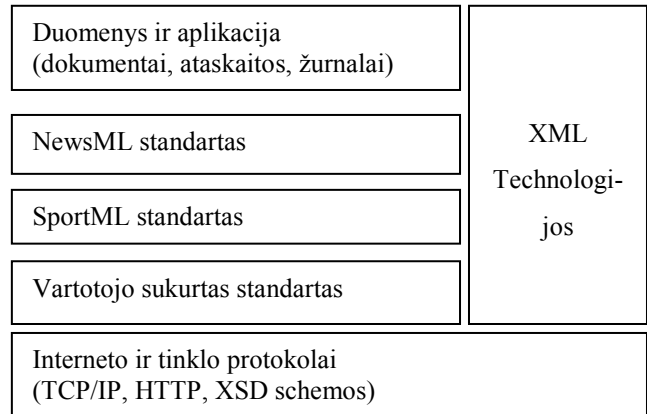
Duomenys SQL serveryje laikomi standartizuotuose XML dokumentuose, tačiau jų loginės struktūros, turi laikytis esminių taikomų XML dokumentų principų.



10 pav. Reikalavimai duomenims

3.4. XML dokumentų naudojami standartai

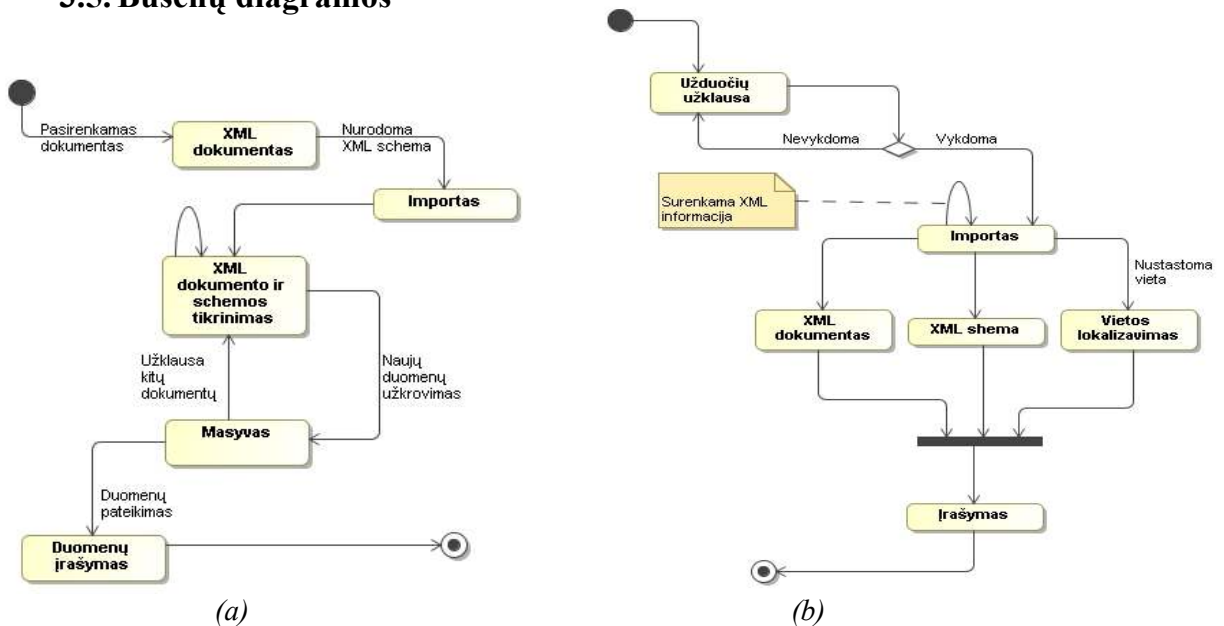
Šiam projekto įgyvendinimui buvo pasirinkti rinkiniai standartų (SportsML, WeatherML, FinML, NewsML), kad būtų lengviau įsisavinti, pritaikyti realius duomenis ir palyginti rezultatus. Standartai yra sukurti ir paremti interneto tinklo duomenų apsikeitimui. Standartai pasirinkti dėl jų nelicencijavimo, be to tai yra atviro kodo – XML technologijos dokumentai, bei gausi jų pasirinkimo galimybė, kuriuos yra plačiai paplitusios visame tinkle (olimpiados, meteorologijos stotys, . Šioje diagramoje pavaizduota, kaip standartai tarpusavyje yra susiję ir sudaro esamą infrastruktūrą.



11 pav. Standartai

Numatyti standartai bus integruoti programinėje įrangoje ir bus vystomi remiantis egzistuojančiais interneto ir tinklo protokolais, bei servisais. Todėl reikia sukurti Web servisą, kuris būtų atsakingas už duomenų migraciją.

3.5. Būsenų diagramos



12 pav. rankinio ir (b) automatizuoto XML dokumento duomenų importas

3.6. Išdėstymo vaizdas

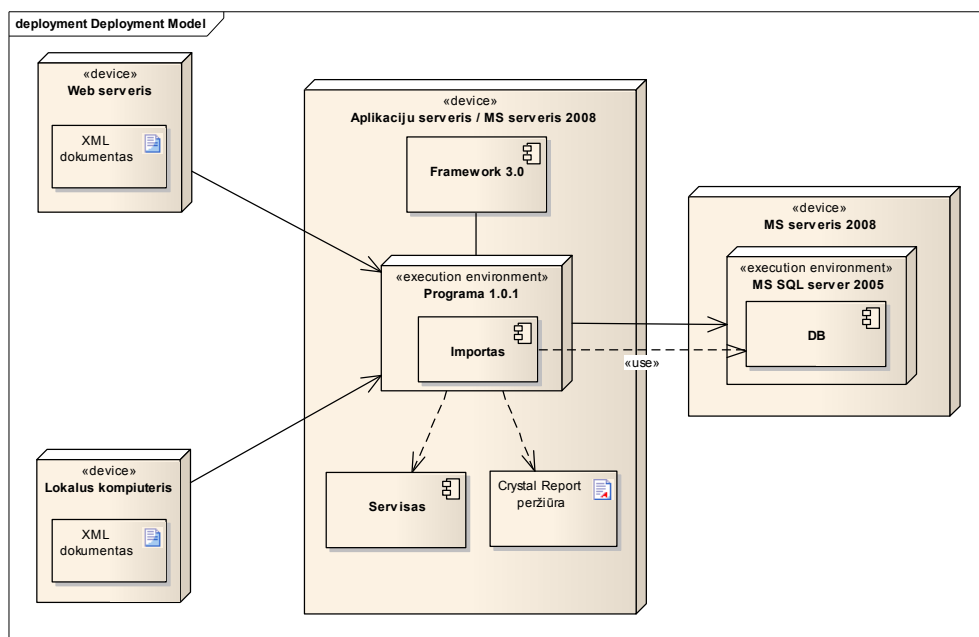
Komponentus ir techninę įrangą paskirstyčiau į keturias grupes. Pirmoji yra WEB serveris, tai nutolęs kompiuteris, kuris pasiekiamas internetu iš kurio galima pasiimti per HTTP protokolą XML dokumento duomenis. WEB serverį galima pakeisti ir lokaliu kompiuteriu, tai yra vietiniame tinkle esantį kompiuterį, kuriame yra įrašytas XML dokumentas arba vartotojas jungiantis prie programos nuotoliniu būdu. Turint duomenis, reikalinga programinė įranga kuri pasiimtu šiuos duomenis. Siekdami didesnio patikimumo ir saugomo naudosime du atskirtus serverius, kur vienas iš jų bus skirtas aplikacijai, o kitas duomenų bazei saugoti. Aplikacijų serveryje instaliuota programinė įranga skirta valdyti duomenims, tai yra migracijai ir priskirti naujas užduotis OS servisui, tau yra automatizuoti (angl. Sheduler) programinę įrangą, be vartotojo įsikišimo. Duomenų bazei laikyti naudojamas MS SQL Server 2005 serveris.

Kaip aptarta ankstesniuose skyriuose, techniniuose architektūros sprendimuose, buvo priimta naudoti kūrimo procese šias technologijas:

- Windows 2008 Server
 - MS SQL Server 2005
- Microsoft .NET technologijos
 - Visual Studio .NET
 - SQL Data provider
 - XML data reader, writer
- Crystal Report for Visual Studio .NET

Šios technologijos paveikia tiek paviršutinius, tiek ir vidinius programinės įrangos sistemos sluoksnius. Tai atspindi šiame grafiniame paveikslėlyje, kuriame matoma kaip kuri sistemos dalis sąveikauja ir yra atsakinga už savo dalies aprūpinimą. Todėl siekiant sukurti programinę įrangą patikimą ir saugią būtina laikytis šios architektūros, išskaidant į šiuos erdvės sluoksnius. Tokiu būdu sėkmingai įgyvendinant programinėje įrangoje.

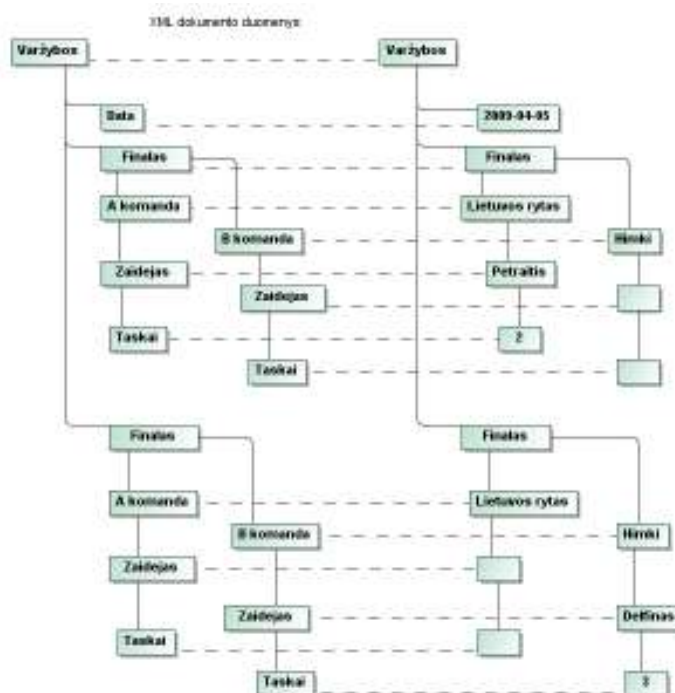
UML išdėstymo modelis pateiktas 13 paveiksle. Be to, apie tai, kokia svarbi architektūra ir jos sprendimai, buvo aptarta anksčiau, šis modelis atitinka ir remia sluoksnių dizaino modelį, kuris aptartas šiame skyriuje.



13 pav. Komponentų ir techninės įrangos sluoksniai

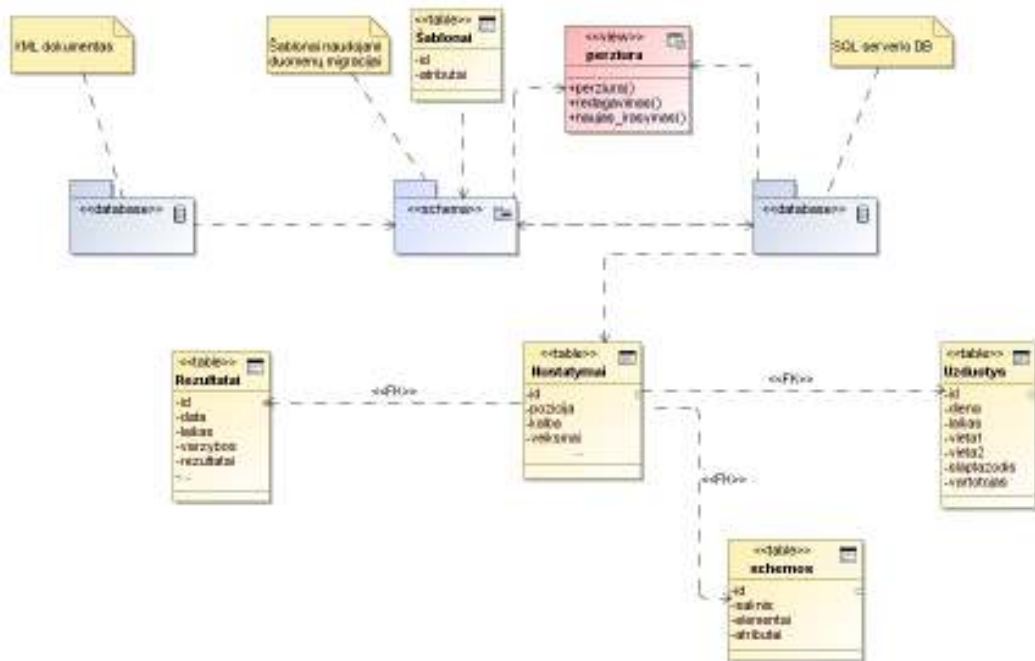
3.7. XML duomenų vaizdas

Paveikslėlyje pateikta iliustracija, kaip atrodytų duomenų nuskaitymas iš XML dokumento į programos atmintį ir į MS SQL serverį. Aišku pradžioje reiktų turėti sudarytą programos duomenų migracijos šabloną. Pavyzdėlyje pavaizduota krepšinio varžybų rezultatai, tačiau ši programinė įranga yra universali ir galima praplėsti, taip pritaikyti kitai pasirinktai sporto šakai ar veiklos sričiai (pvz. orai, temperatūra, prognozės) sukuriant savo migracijai skirtą šabloną.



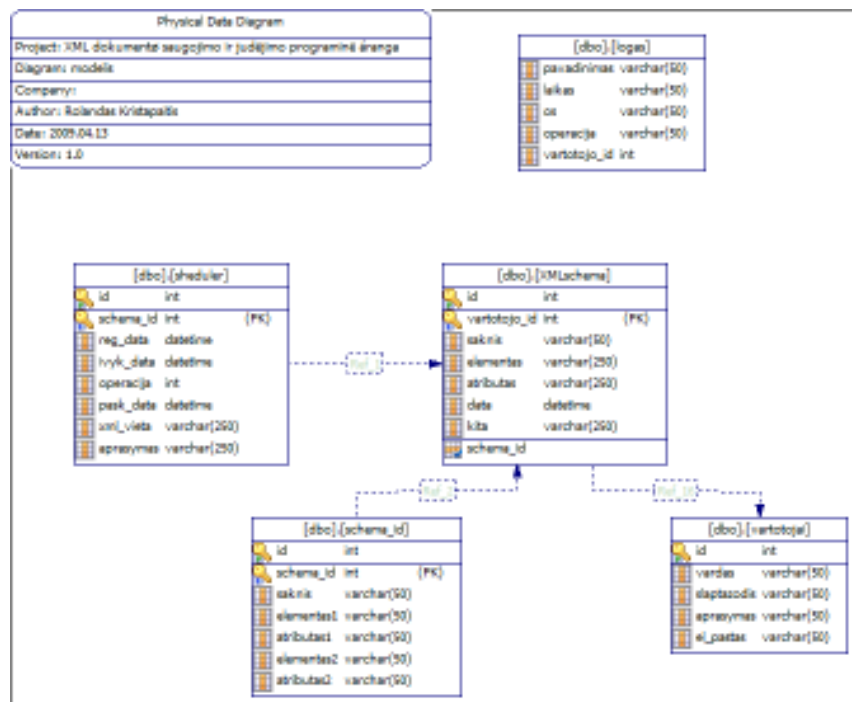
14 pav. Vartotojo sukurtos XML schemas pavyzdys su duomenimis

XML dokumento ir MS SQL serverio ryšys su esančia DB (schemas detalumas priklauso nuo pasirinktos veiklos srities) lieka nepakitęs, keičiasi tik fizinė DB, kuri pavaizduota žemiau paveikslėlyje.

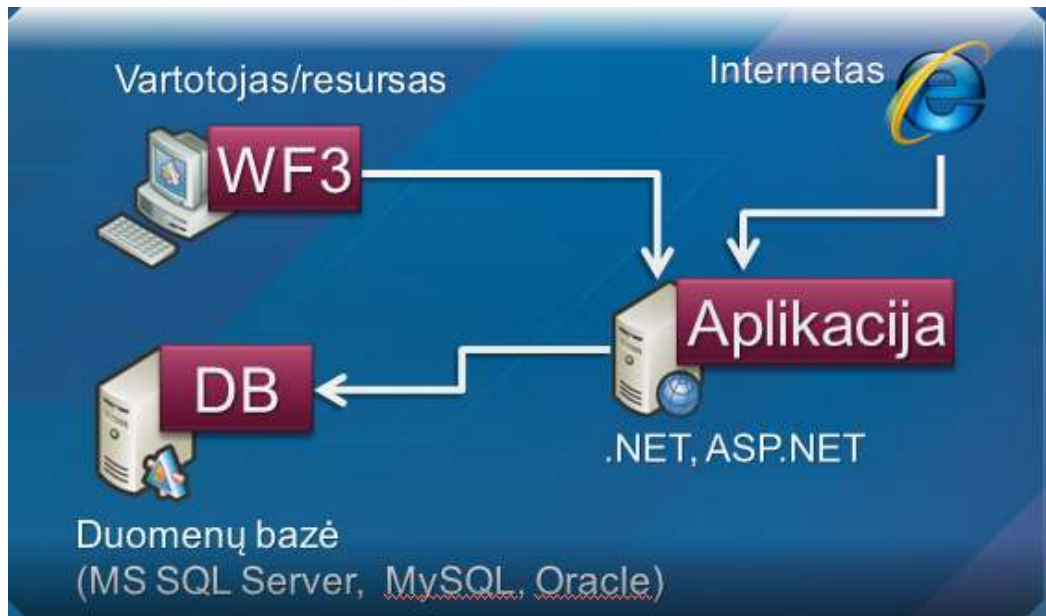


15 pav. XML ir MS SQL migracijai skirta architektūra

Žemiau esančiame 14 paveikslėlyje pateiktas DB modelis, kuriame pavaizduota tik vienos XML schemas lentelė, o jos duomenys saugomi lentelėje (*schema_id*), todėl plečiant ir kuriant kitas XML schemas automatiškai auga kartu ir DB architektūra.



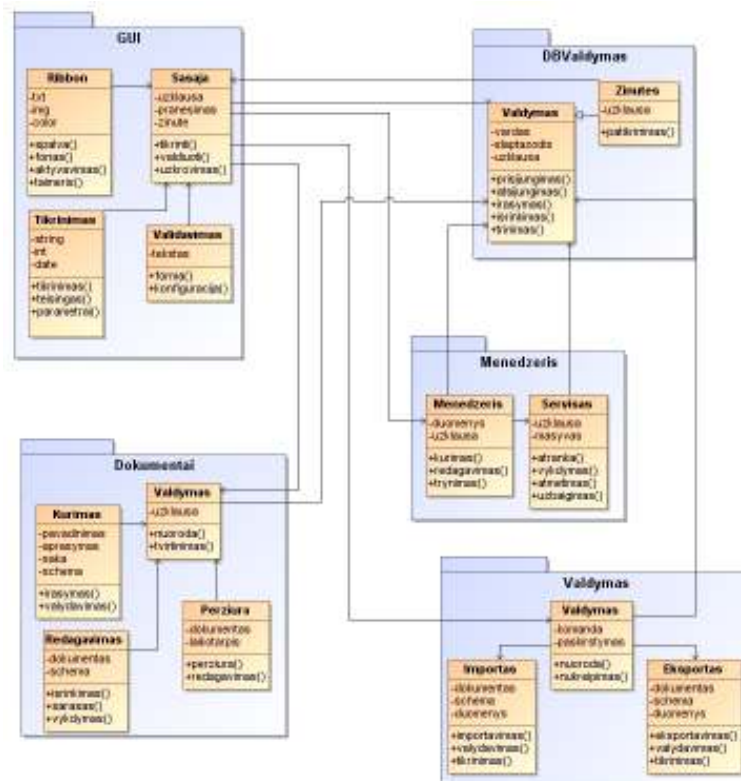
16 pav. MS SQL duomenų bazės vaizdas



17 pav. Duomenų srautų diagrama

3.8. Aplikacijos paketai

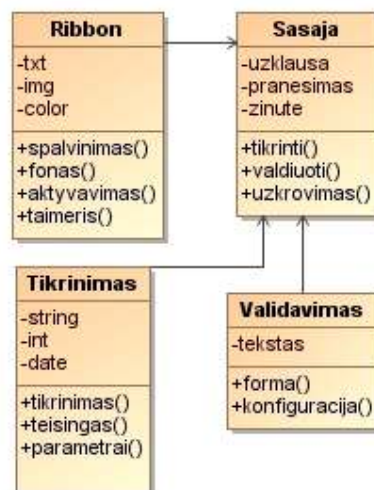
PĮ struktūra sudaryta iš pagrindinio paketo, kuris sudarytas iš žemesniame lygyje esančių penkių komponentų: GUI, Dokumentai, Valdymas, Menedžeris, DB Valdymas. Juos aptarsime žemiau. Tai pagrindinis paketas, kuris užtikrina pagrindinį programos veikimo principą. Toliau aprašomi pakete esantys komponentai ir jų taikymo paskirtis.



18 pav. Aplikacijos paketai

3.8.1. GUI komponentas

- Klasifikacija. *GUI* Komponentas.
- Apibrėžimas. Komponentas skirtas valdyti grafinę vartotojo sąsają.
- Atsakomybės. Pagrindinė šio komponento atsakomybė yra grafinių elementų paskirstymas ir jų įvedimo informacijos tikrinimas ir validavimas. Grafinė vartotojo sąsaja atsakinga už kiekvienos funkcijos – lango, pranešimų iškvietimą ir komandų iškvietimą. Tai komponentas, kuris sujungia vartotojo veiksmus su tiesioginėmis kitų komponentų funkcijomis (dokumentų kūrimas, importas, eksportas ir kt.).
- Apribojimai. Turi turėti ryšį su *DBValdymo*, *Dokumentai*, *Menedzerio*, *Valdymas* komponentais.
- Struktūra. Šio komponento viduje yra keturios pagrindinės klasės: *Ribbon*, *Sąsaja*, *Tikrinimas* ir *Validavimas*.
- Sąveikavimas. Kaip ir buvo rašyta prieš tai, GUI komponentas glaudžiai bendradarbiauja su kitais komponentais: *DBValdymas*, *Dokumentai*, *Menedzeris* ir *Valdymas*. Programinė įranga taip vykdo ir iššaukia naujas komandas, užklausas, kurios atsakingos už duomenų įrašymą, išrinkimą ir vykdymą.
- Resursai. Šiam komponentui reikalinga .NET Framework 3.0 versija. Apie 150MB vietos.

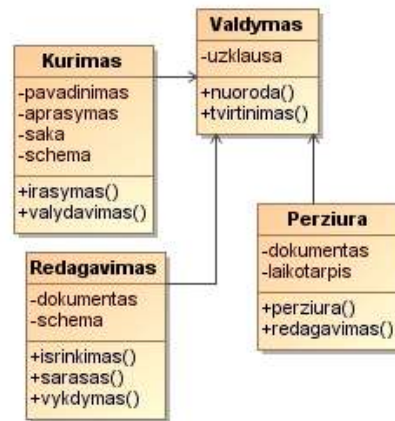


19 pav. GUI klasės

3.8.2. Dokumentų valdymo komponentas

- Klasifikacija. Komponentas.
- Apibrėžimas. Komponento paskirtis sukurti naują XML dokumentą ir paruošti jį duomenų migravimui.
- Atsakomybės. Kuria XML dokumentus, leidžia juos redaguoti, bei peržiūrėti jų struktūrą ir turinį.

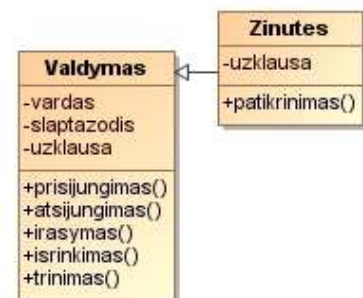
- Struktūra. Komponento struktūra pateikta 4 paveikslėlyje.
- Sąveikavimas. Bendradarbiauja su GUI, DBValdymo komponentais.
- Sąsaja/eksportas. Sudaro keturios pagrindinės klasės.



20 pav. Dokumentų valdymo komponentas ir jo klasės

3.8.3. DBValdymas komponentas

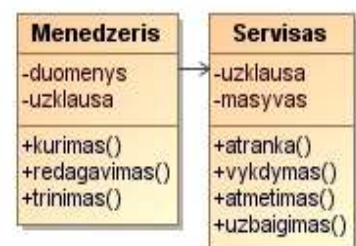
1. Klasifikacija. Komponentas.
2. Apibrėžimas. Esant kelioms RDBVS, reikalingas ir specialus DB valdymo komponentas.
3. Atsakomybės. Vykdo užklausas, kurios atsakingos už XML dokumentų duomenų migraciją.
4. Apribojimai. Turi būti ryšys su bent viena DB.
5. Struktūra. Sudaryta iš dviejų klasių, kurios atsakingos už DB valdymą. Struktūra pateikta
6. Sąveikavimas. Glaudžiai sąveikauja su kitomis PĮ komponentų dalimis.
7. Resursai. NET Framework 3.0 duomenų šaltinių palaikymo bibliotekos: MssqlAdapter, OracleDevelopAdapter.



21 pav. DBValdymo komponentas ir jo klasės

3.8.4. Menedžerio komponentas

- Klasifikacija. Modulis.
- Apibrėžimas. Skirtas automatizuoti sistemą ir užsakyti vykdomas užduotis.
- Atsakomybės. Turi būti pilnai užpildyta užduotis su egzistuojančiu dokumentu ir jame esančiais duomenimis.



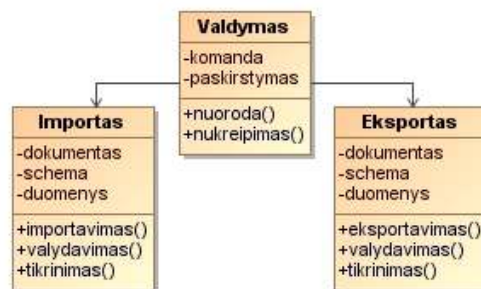
22 pav. Menedžerio komponentas ir jo klasės

- Apribojimai. Vartotojas pats registruodamas naują užduoti pasirenka vykdymo laiką ir operacijos rūšį.
- Struktūra. Paveikslėlyje pavaizduotos dvi klasės ir metodai juose.
- Sąveikavimas. Šis modulis bendradarbiauja su DBValdymas ir GUI komponentais.
- Resursai: .NET Framework 3.0 versija.

3.8.5. Valdymo komponentas

- Klasifikacija. Paketas.
- Apibrėžimas. Paketo tikslas valdyti XML dokumentų duomenų srautus.
- Atsakomybės. Migruoja duomenys iš vienos pasirinktos DB į kitą.
- Apribojimai. nėra
- Struktūra. Pateikta 8 paveiksle.
- Sąveikavimas. Bendradarbiauja su GUI sąsajos klasėmis, DBValdymo komponentu.
- Resursai. .NET Framework 3.0 versija.

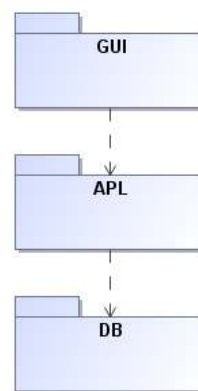
- Skaičiavimai. -
8. Sąsaja/eksportas. XmlParserContext, XmlAttribute, XmlElement, XmlDocument, XmlNode, XmlReader, XmlUrlResolver, XmlTextReader.



23 pav. Valdymo komponentas ir jo klasės

3.9. Sistemos statinis vaizdas

Sistemos statinis vaizdas yra vienas iš sudėtingiausių ir daugiausiai laiko užimantis sprendimo uždavinys, sprendžiant vieną ar kitą pasirinkimo variantą. Todėl didžiausias dėmesys projektavimo dėmesys skiriamas architektūrai. Visų pirma aprašoma struktūrinė sistemos hierarchija, kuri susideda iš sluoksnių, paketų ir posistemių. Tada apibūdiname kaip šie paketai bendradarbiauja tarpusavyje ir suteikia didesnę architektūros funkcionalumą.

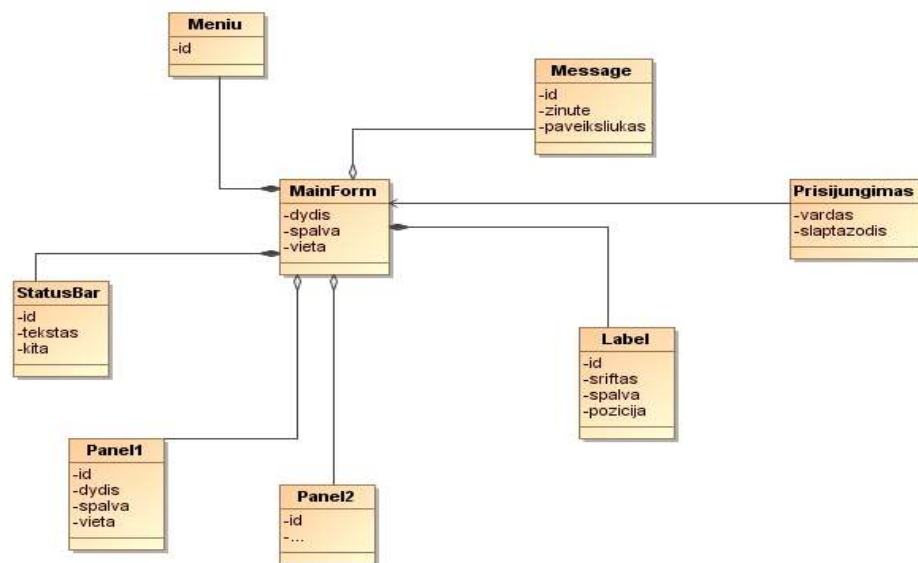


24 pav. Sistemos paketų grafinis vaizdas

3.10. Paketų detalizavimas

Bendrai taikant sluoksnių architektūra yra apibrėžiami svarbiausi objektai, iš kurių kiekvienam priskiriama atsakomybė už tam tikrą darbą. Sluoksniuota hierarchinė struktūra izoliuoja ir organizuoja įvairias sistemos prieigas, ji sudaryta iš šių trijų sluoksnių (žr. 5 paveikslą):

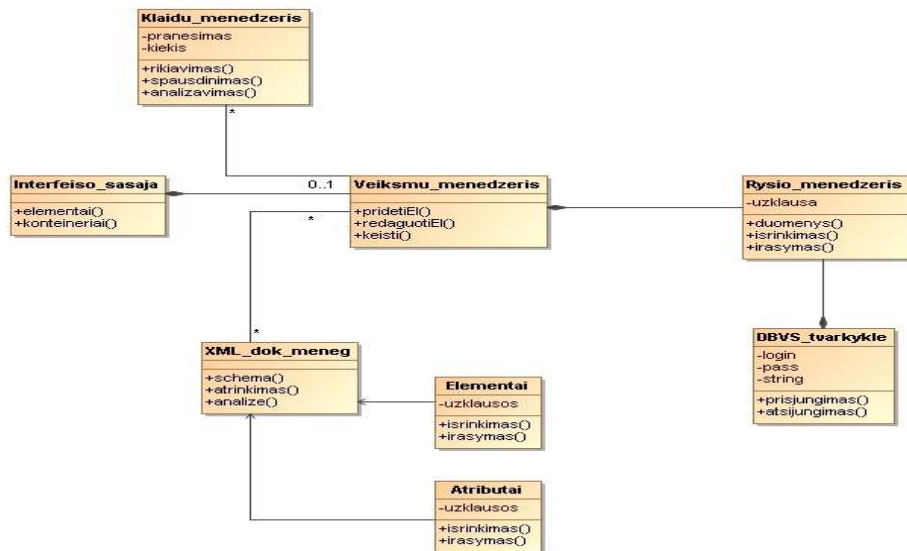
- **GUI sluoksnis.** Šis viršutinis sluoksnis teikia pagalbos sąveiką tarp dalyvių ar vartotojų, sistemos ir programinė įranga, sistema funkcionalumas pristatomas per grafinę vartotojo sąsają.



25 pav. GUI klasės

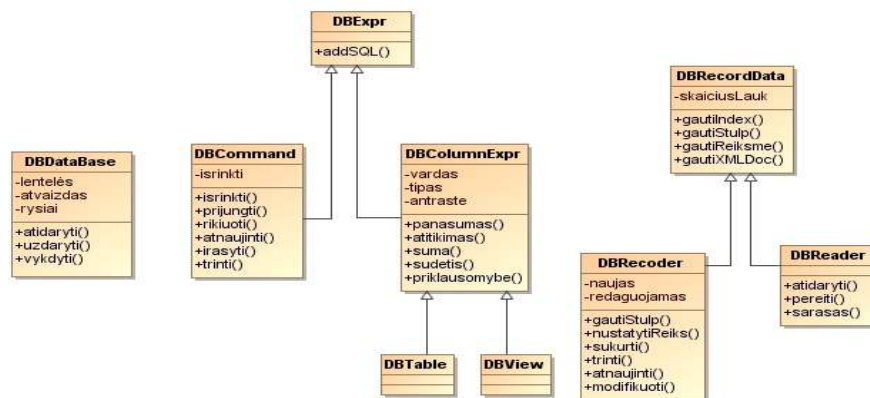
Vartotojas prisiregistravęs prie sistemos gauna priėjimą prie RDBMS, kitaip tariant prie duomenų bazės. StatusBar klasė skirta pateikti svarbią informaciją, prisijungimo laiką, vartotojo duomenys, operacijos būseną. Panel1, Panel2 – išdėstomi grafiniai programos elementai: įvedimo laukai, mygtukai. Menių lange matosi visa programinės įrangos navigacija (duomenų eksportas, importas, XML schemas, ataskaitos). Message klasė skirta sugeneruoti pranešimui apie sistemos darbą – klaidas, sutrikimus, kita informaciją.

- **Aplikacijos logikos sluoksnis.** Šis vidurinis sluoksnis suteikia paramą taikant konkrečius procesus, veiksmus, kurie nepriskiriami prie DB ir GUI sluoksnių. Tai papildomos loginės operacinės, funkcijos, kurios besąlygiškai bendradarbiauja su duomenų baze.



26 pav. Aplikacijos pagrindinės klasės

- **Duomenų prieigos sluoksnis.** Šis apatinis sluoksnis suteikia prieigą prie duomenų ir jų panaudojamumą. Tai reliacinė duomenų bazė lygis.



27 pav. DB klasės

Dvi pagrindinės klasės (DBExpr ir DBRecordData), kurios atsakingos už duomenų įrašymą ir nuskaitymą. Kitos hierarchinės klasės jungiasi į antrą lygį ir pasiskirsto pagal paskirtį. DBComand klasė atsakinga už SQL užklausą. DBColumnExpr atrenka duomenų bazės stulpelius ir pateikia grafišką vaizdą.

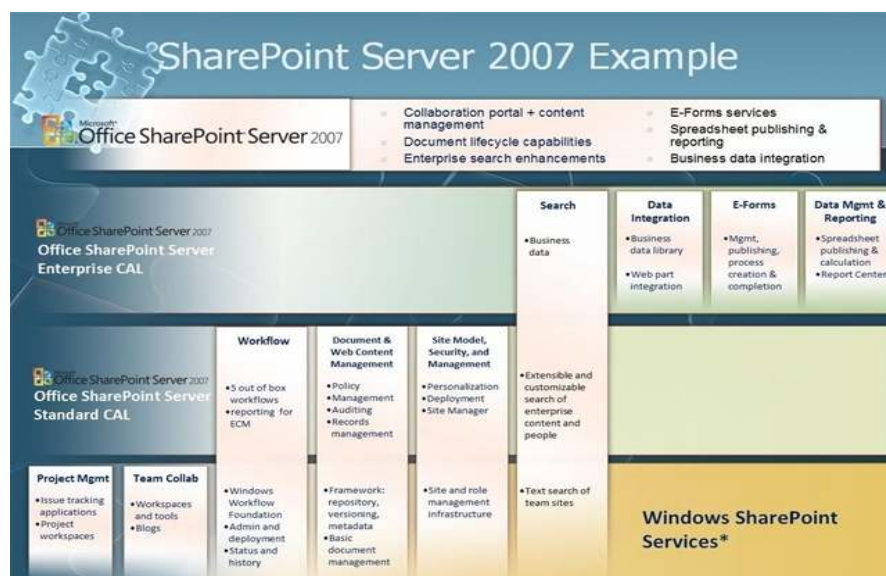
PĮ architektūra pagrįsta sluoksniais, todėl egzistuoja vadinama hierarchinė struktūra, kuri sako kad kiekvienas sluoksnis turi turėti ryšį su kitu esančiu po jo sluoksniu. Tarkim duomenų sluoksnis, kuris atsakingas už duomenų paskirstymą, negali funkcionuoti, jeigu neturi ryšio su aplikacijos sluoksniu, kuris atsakingas už procesų ir veiksmų paskirstymą, lygiai taip pat negali pilnai funkcionuoti ir aplikacijos sluoksnis, jeigu nėra ryšio su grafine vartoto sąsaja. Tačiau egzistuoja tokia galimybė, kad ne visada visi sluoksniai turi ryšį. Sistema gali funkcionuoti automatizuotai ir be GUI sluoksnio, valdant programos operacijas servaisais.

4. PROGRAMINĖS ĮRANGOS TYRIMO DALIS

Tyrimo metu buvo siekiama ištirti esamą situaciją duomenų apsikeitimo tarpe, bei išrinkti iš įvairios panašaus pobūdžio programinės įrangos siūlomas ir esamas funkcijas, bei jas analogiškai perkelti į kuriamą sistemą. Apskritai, kalbant apie tokios specializacijos programinius įrankius, galiu teigti, kad elementaraus konkurento nėra, kuris galėtų pilnai valdyti XML dokumentus. Ieškant panašių produktų galima paminėti kelis žinomesnius⁷, kurie patraukė autoriaus dėmesį, tačiau jų universalumas ir negalima pasigirti gausa, o tuo labiau ieškant nemokamos programinės įrangos. Sakoma, kad norint pasiekti ir susilaukti gerų rezultatų, atsiliepiamų reikia griebti ne kuo daugiau, o kokybiškiau (Scot Hiller, 2007). Nieko įprasto, kai daugelis programinių įrangų funkcijų yra nereikalingos arba visiškai pasenusios ir nebeaudojamos, todėl kam to reikia vartotojui, o tuo labiau, jeigu PĮ yra mokama.

4.1. Dokumentų apsikeitimas

Vienas iš patogių sprendimų, kaip apsikeisti svarbia informacija pasiūlytas Microsoft sprendimas. IT rinkoje egzistuoja PĮ pavadinimu „SharePoint“, tai Microsoft sukurtos platformos verslui versija, skirta bendrai naudoti informaciją, dirbti komandoje, kurti ir dalintis dokumentais, o taip pat valdyti bendruomenių ir asmenų vykdomus procesus. Šios platformos pagalba dokumentus galima kurti ir redaguoti vienu metu „gyvai“. Sukūrus reikiamą dokumentą ir suteikus prieigą pasirinktiems bendradarbiams, dokumentas „pakabinamas“ virtualioje erdvėje, tad su juo gali dirbti keli žmonės vienu metu - taip svarbių dokumentų kūrimas vyksta žymiai sparčiau, nes visi darbuotojai savo kompetenciją gali pritaikyti iškart.

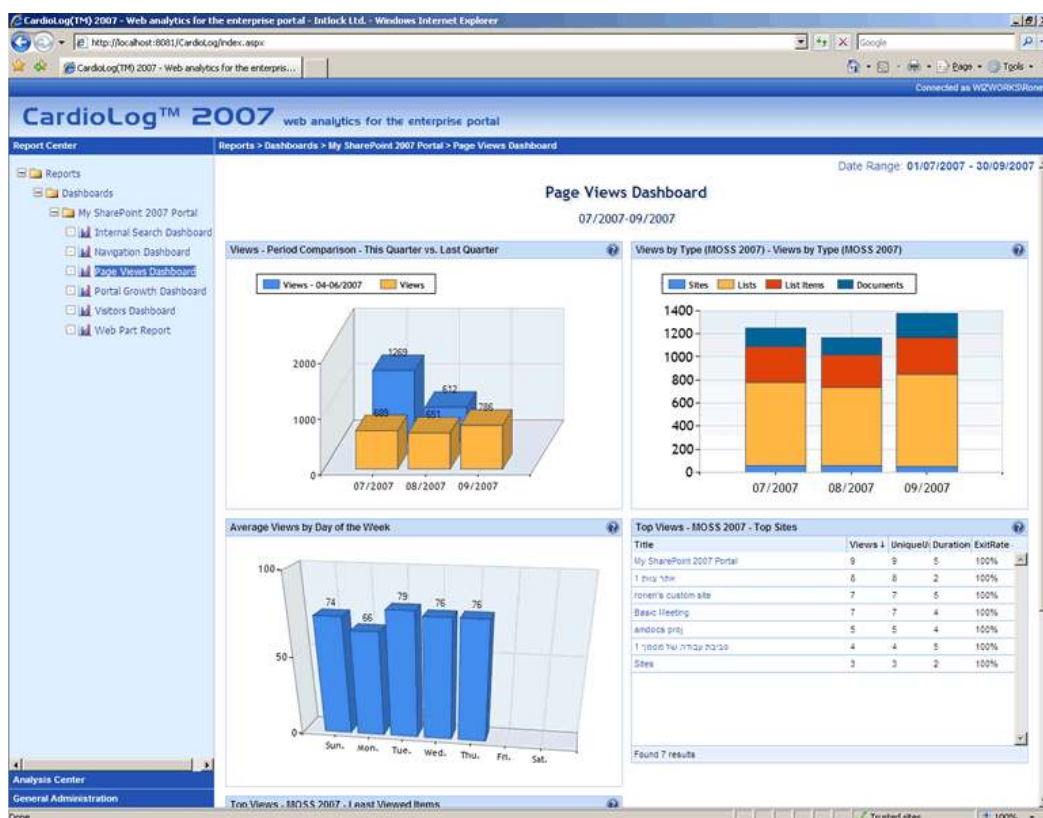


28 pav. SharePoint paslaugos

⁷ SharePoint - <http://sharepoint.microsoft.com/en-us/Pages/default.aspx>

Microsoft LAR (Large Account Reseller) statusą turinčios tarptautinės IT bendrovės „DPA programinė įranga“ paslaugų skyriaus vadovas Egidijus Antanaitis teigia, kad atsižvelgiant į dabartinės darbo ir bendravimo virtualioje aplinkoje tendencijas, diktuojamas tokių socialinių tinklų, kaip „Facebook“ ir „LinkedIn“, naujausia „SharePoint 2010“ platforma verslo vartotojams siūlo įvairias užduotis atlikti ir vidinę komunikaciją vykdyti socialinių tinklų principu. „SharePoint 2010“ vartotojo sąsaja vizualiai panaši į šiuo metu populiarių socialinių tinklų struktūrinį išdėstymą - platformą naudojantys darbuotojai gali nuolat keisti savo statusą, skelbti naujienas ar dalintis įmonės internetinės svetainės turiniu“, - pasakoja „DPA programinė įranga“ paslaugų skyriaus vadovas⁸.

Sugeneruotas puslapis – ataskaita:



29 pav. MS SharePoint ataskaita

Tokios PĮ dėka yra supaprastintas informacijos rinkimas, padidintas informacijos saugumas, pasitelkiant kompleksines priemones, skirtas elektroniniam turiniui tvarkyti ir kontroliuoti. Racionalizuojamas organizacijos našumas, mažinamas kasdieninis procesas naudojantis elektroninėmis formomis ir savaiminius darbo eigos procesus, kuriuos vartotojai gali inicijuoti, sekti ir juose dalyvauti naudodami įprastines „Microsoft Office“ programas, el. pašta ar interneto naršykles (John Holliday, John Alexander, 2007).

- **Kontroliuojami dokumentai pasitelkus išsamaus ir išplėstinio valdymo strategiją.** Nustatoma individualizuota dokumentų valdymo strategija, kad galima būtų kontroliuoti prieigos teises atskirų elementų lygmeniu, apibrėžiamo saugojimo periodai bei veiksmai jam pasibaigus. Strategijos integravimas su įprastomis taikomosiomis programomis padaro pavedimų procesus skaidrius ir lengvus naudoti vartotojams. Integravimas su informacijos teisių valdymu padeda užtikrinti geresnę privačios ir konfidencialios informacijos apsaugą, net jei ji neprijungta prie serverio.
- **Centralizuotas saugojimas, tvarkymas ir pasiekimas dokumentų.** Organizacijos gali saugoti ir tvarkyti visus verslo dokumentus ir turinį vienoje vietoje, o vartotojai turi mechanizmą, leidžiantį naršyti ir surasti reikiamą informaciją. Numatytuosius saugyklos parametrus galima modifikuoti ir pridėti prie darbo eigos, apibrėžti laikymo strategiją ir pridėti naujų šablonų bei įvairių rūšių turinio.
- **Supaprastintas žiniatinklio turinio valdymas.** Suteikia nesudėtingas funkcijas, leidžiančias kurti, patvirtinti ir skelbti žiniatinklio turinį. Pagrindiniai puslapiai ir puslapių maketai pateikia šablonus, kuriuos galima pakartotinai naudoti ir kurie užtikrina nuoseklų vaizdą ir nuotaiką. Naujos funkcijos leidžia įmonėms skelbti turinį iš vienos srities į kitą (pavyzdžiui, iš bendrų tinklalapių į portalą) arba ekonomiškai tvarkyti daugiakalbio turinio skelbimą intranete, ektranete ir interneto puslapiuose.
- **Išplečiamas procesas už organizacijos ribų.** Formų paslaugomis paremti sprendimai leidžia saugiau ir tiksliau rinkti informaciją tiek organizacijos viduje, tiek išorėje, nekoduojant jokių individualizuotų taikomųjų programų. Ši informacija gali būti lengvai integruojama į specialias verslo sistemas, išsaugoma dokumentų bibliotekose, naudojama darbo eigos procesams pradėti arba skelbiama žiniatinklio tarnybose, taip išvengiant dvigubų pastangų ir brangių klaidų, galinčių atsirasti dėl duomenų įvedimo rankiniu būdu.
- **Racionalizavimas kasdienės veiklos.** Naudojantis darbo procesais galima automatizuoti ir padaryti užduotis, pavyzdžiui, dokumentų peržiūrą ir tvirtinimą, problemų sekimą ir parašų rinkimą. Integravimas su įprastinėmis taikomosiomis „Microsoft Office“ programomis, el. pašto programomis ir interneto naršyklėmis supaprastina vartotojų darbą. Organizacijos gali lengvai keisti savaiminius procesus arba nustatyti nuosavus procesus.

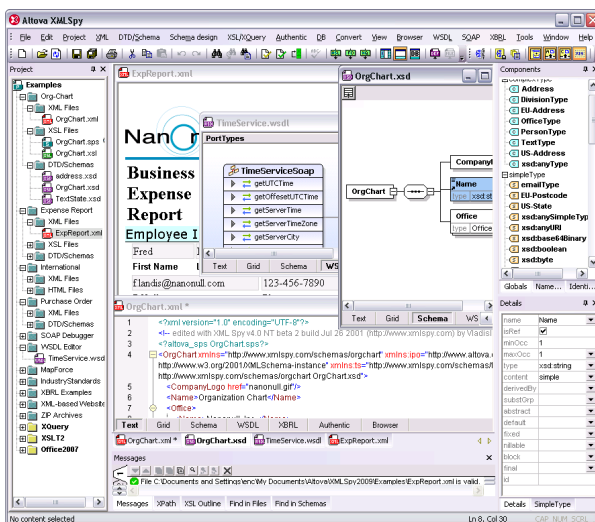


30 pav. MS SharePoint funkcionalumas

Naudojantis tokia PL organizacijos gali itin operatyviai reaguoti į klientų pageidavimus tiesiog iš savo darbo vietas – taip sumažėja keliavimo poreikis, o kartu ir to sąnaudos. „SharePoint“ platforma užtikrina ir visos turimos elektroninės informacijos centralizaciją. Nesvarbu, ar organizacija turi visoje šalyje išsidėsčiusius padalinius, ar visi darbuotojai dirba viename pastate. Visa elektroninėse laikmenose esanti informacija gali būti apjungta į vieningą duomenų bazę, kuri ne tik apsaugota patikima antivirusine programa, bet ir integruota į rezervinių duomenų kopijų kūrimo sistemą. Taip visos turimos informacijos išeiga ir įeiga yra aiškiai kontroliuojama, galima reguliuoti ir informacijos prieinamumą.

4.2. XML dokumentų analizės įrankis

Visas programines įrangas sieja vienas bendras dalykas – koncentravimasis į objektą. Labai geras pavyzdys siejamas įrankis Altova, kuriame integruotas komplektas priemonių padedančių dirbti su reliacinėmis duomenų bazėmis, XML ir UML. AltovaSpy puiki priemonė skirta kurti ir redaguoti XML dokumentus, schemas (DTD, XSD).

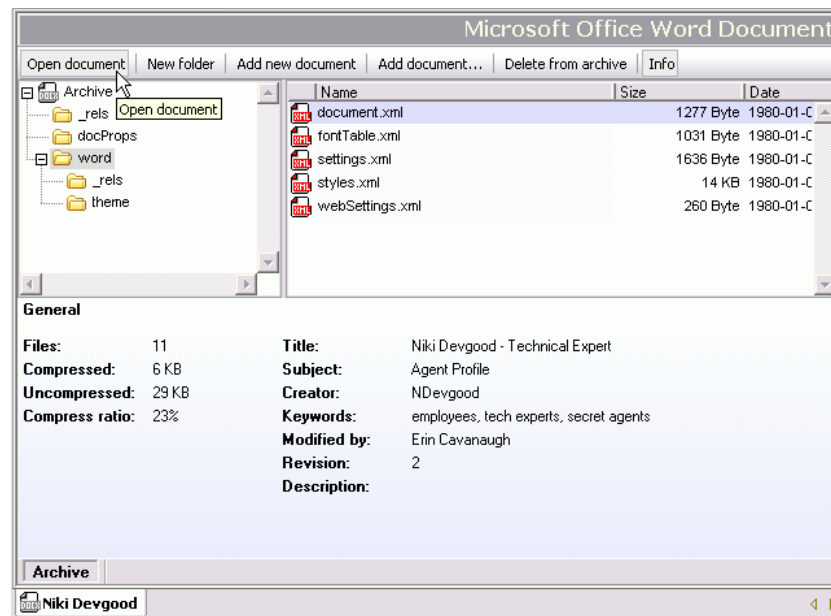


31 pav. AltovaSpy GUI

¹⁰ Verslo naujienos, http://vz.lt/straipsnis/2010/05/13/Microsoft_SharePoint_2010_verslui_pades_iveikti_infor, [žiūrėta 2010-04-13]

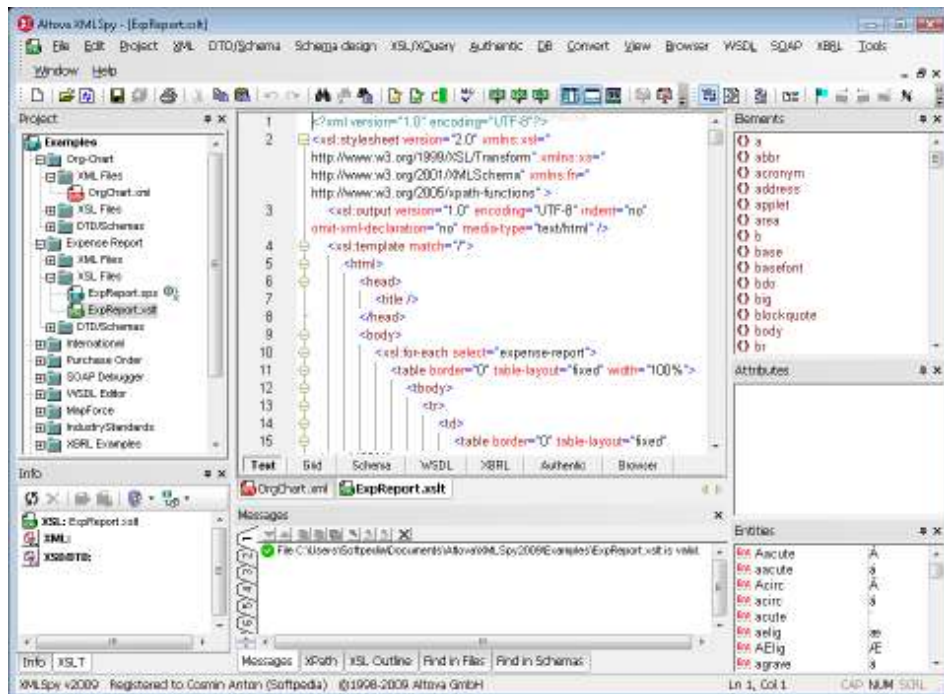
¹¹ Altova XML valdymo įrankio video pristatymas, <http://www.altova.com/videos.asp>, [žiūrėta 2010-04-13]

Įrankio pagalba, galima sekti dokumentų pakeitimus, jų atributus (antraštes, autorių, raktinius žodžius, peržiūros, taisymo, sukūrimo laikas).



32 pav. Dokumentų tvarkymas

Grafiniame įrankio redaktoriuje patogiai išdėstomi laukai (elementai, atributai, projekto dokumentai, pranešimai).



33 pav. AltovaSpy XML dokumento redaktorių

18 lentelė. Programinės įrangos savybių palyginimas

Funkcijos	SharePoint	AltovaSpy	Moorena
Dokumentų teisės	+	-	+
Autorizacija	+	-	+
XML dokumentų redaktorius	-	+	+
Centralizuotas duomenų priėjimas	+	-	+
Grafinis XML redaktorius	-	+	-
XML validacija	-	+	+
Duomenų bazių integravimas	+	+	+
XPath analizatorius	-	+	+
DTD schemas konverteris	-	+	-
Java/C#/C++ kodo generavimas	-	+	-
XML projektų valdymas	+	+	+
Užduočių paskirstymas	+	-	+
Komunikavimo priemonės (paštas, žurnalas)	+	-	+
Nemokama programinė įranga	-	- (30d. Bandomasis)	+

4.3. Dokumentų pildymo formos

Kadangi jau buvo šnekėta apie XML dokumentų privalumus besikeičiant svarbia informacija, kyla natūralus klausimas dėl formų pildymo. Kaip galima paprastai XML dokumentus pildyti, kad nereiktų išmanyti schemų. Toks sprendimas yra realizuotas kai kuriuose įrankiuose (InfoPath, XForms). Šios formos leidžia pildyti ir tvarkyti duomenis patogiu vartotojui būdu, nes dalis informacijos, kuri yra nebūtina (XSD, DTD, žymės) paslepiama po šablono forma. Be to, InfoPath siūlo

The image shows a web form interface. At the top, there is a banner with a woman's photo and the text 'Welcome to Fasthelp'. Below this, the form is divided into two main sections: 'Problem Details' and 'User Information'. The 'Problem Details' section includes a 'Problem Title' text input field, a 'Problem Category' dropdown menu (with a list of categories: Communications Tools, Hardware, Network Access, Other, Software), and a 'Sub-Category' dropdown menu. The 'User Information' section includes fields for 'Logged By', 'Full Name', 'E-mail', 'Phone', and a 'Location' dropdown menu.

neblogą integraciją su SharePoint produktu, kuris pagyviną ir papildo šių galimybių panaudojimą.

34 pav. Dokumentų pildymo formos

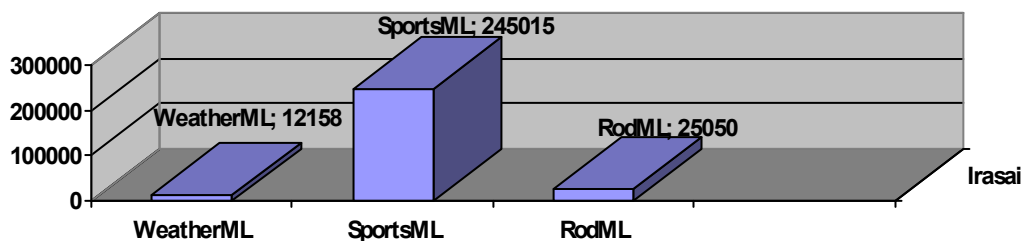
5. PROGRAMINĖS ĮRANGOS EKSPERIMENTINIAI

5.1. Eksperimentiniai duomenys

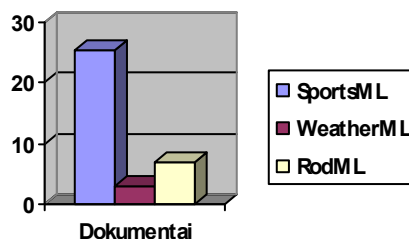
Eksperimento metu buvo panaudoti atsitiktiniai trys skirtingų tipų ir dydžių XML dokumentų standartai. Du buvo paimti globalūs SportsML¹², WeatherML¹³ ir kitas pasirašytas autoriaus – RodML¹⁴. WeatherML yra skirtas orų prognozėms kaupti, SportsML standartas skirtas sporto duomenims rinkti, o RodML – gamybos duomenims (skaitiklių parodymai) sekti. Lentelėje (Nr. 19) pateikiami standartų pavadinimai, dokumentuose esantis įrašų skaičius, iš kur buvo paimti duomenys, standartai ir dokumento sudėtingumo lygis. Lygis nustatytas pagal XML dokumentų struktūrą ir išdėstymo vaizdą.

19 lentelė. Eksperimento duomenys

Standartas	Įrašų sk.	Šaltinis	Sudėtingumas
SportsML	245015	http://www.iptc.org/cms/site/index.html	Aukštas
WeatherML	12158	http://www.wunderground.com/	Žemas
RodML	25050	http://www.pe.lt , http://www.dujos.lt	Vidutinis



35 pav. Dokumentų įrašų skaičius



36 pav. Dokumentų dydžiai, MB

¹² SportsML dokumento standartas: <http://www.iptc.org/cms/site/index.html?channel=CH0105>

¹³ WeatherL dokumento standartas: <http://xml.coverpages.org/weatherML.html>

¹⁴ Palaikančios standartą įmonės: <http://www.pe.lt>, <http://www.lietuvosdujos.lt>

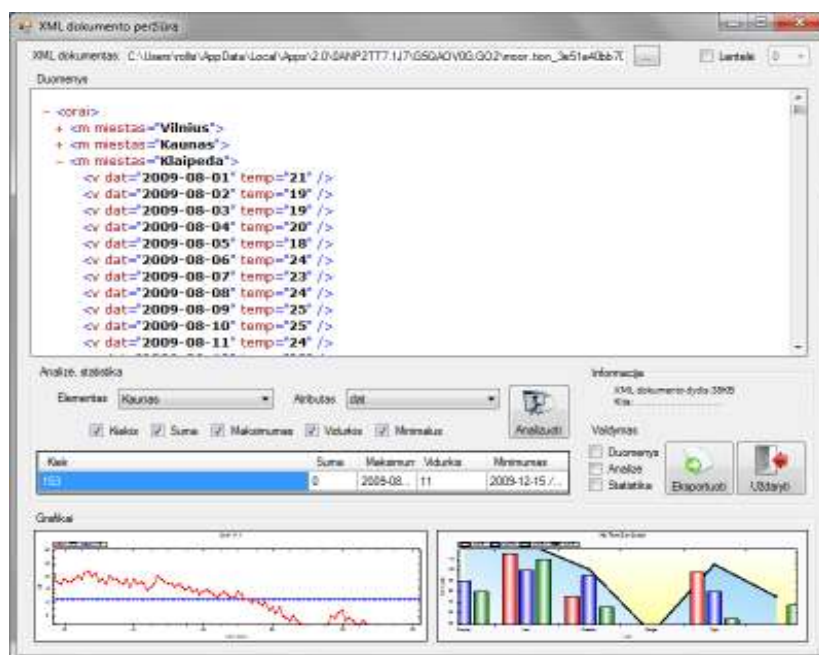
Pastaba. Programinė įranga analizuoja dokumentus tik aprašytais XML dokumentų (DTD, XSD) schemomis.

5.2. Eksperimentų sąlygos ir metrikos

Eksperimentų metu 5.1 skyriuje išvardinti dokumentų standartai ir jų duomenys buvo analizuojami atitinkamai pagal jų vykdymo laikus ir gautus tikslus rezultatus. Gauti rezultatai buvo klasifikuojami pagal sudėtingumą, t.y. taikant elementarius algoritmus (skirtumas, suma, minimumas, maksimumas), bei išrenkant XML dokumento duomenis skirtingame hierarchiniame lygmenyje. Pamėginsime koncentruotis į gautų rezultatų tikslumą ir vykdymo laiką. Priede C pateikiama smulkesnė informacija apie surinktą ir atspausdintą informaciją eksperimento metu. Toliau pateiksime palyginimą su PĮ Altova ir aprašysime gautus rezultatus.

Tam, kad būtų gauti tikslesni rezultatai buvo eksperimentuojama abejomis programomis po keletą kartų, tada skaičiai sulyginti ir pateikti grafikuose. Eksperimento metu buvo imituojamas vartotojų prisijungimas ir lyginamas operacijų vykdymo laikas. Trečiasis eksperimentas, perkeltų dokumentų kiekio ir laiko palyginimas su Microsoft SharePoint programine įranga.

Žemiau pateikta autoriaus sukurtos programinės įrangos analizės modulio dalis skirta duomenų apdorojimui. Diagramose autoriaus sukurta programinė įranga įvardijama kodiniu pavadinimu „Moorena“¹⁵. Detalesnę programos instrukciją rasite priede B.



37 pav. Programinės įrangos analizės įrankis

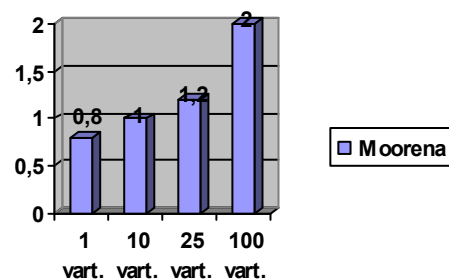
¹⁵ Moorena įrankį galite išbandyti ir parsisiųsti adresu <http://moorena.roleka.lt/>

5.2.1. WeatherML eksperimentas

Dokumentas turi keletą elementų ir po 5 jo atributus. Dokumento informacija išskaidoma į dvi dalis: tekstinė ir skaitinė. Pirmojo eksperimento a) metu buvo imituotas prisijungusių vartotojų skaičius ir stebimi gauti rezultatai.

Eksperimentas a) ir jo savybės:

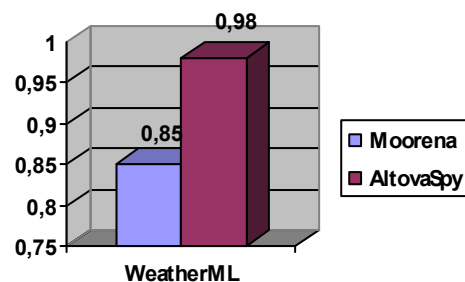
- 1, 10, 25, 100 imituoti vartotojų prisijungimai;
- Interaktyvaus grafiko stebėjimas;
- Temperatūros svyravimai.



38 pav. Eksperimento a) vykdymo laikas (s)

Eksperimentas b) ir jo savybės:

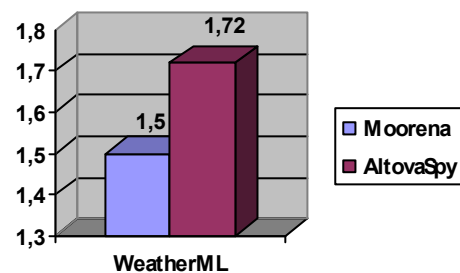
- Vienas vartotojas;
- Vidutinė miestų temperatūra;
- Išrinkta 10 miestų vidutinė temperatūra.
- Rezultatų patikrinimas.



39 pav. Eksperimento b) vykdymo laikas (s)

Eksperimentas c) ir jo savybės:

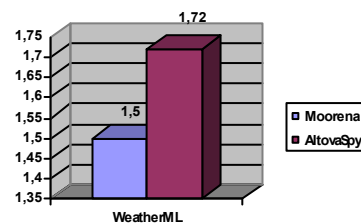
- Laiko intervalas (2009.01.01-2009.09.09);
- Miestas: Kaunas
- Išrinkti duoto laikotarpio temperatūros;
- Temperatūros svyravimai.



40 pav. Eksperimento c) vykdymo laikas (s)

Eksperimentas d) ir jo savybės:

- Duomenų nuskaitymas;
- 50 naujų temperatūrų;
- Validacija ir įrašymas.



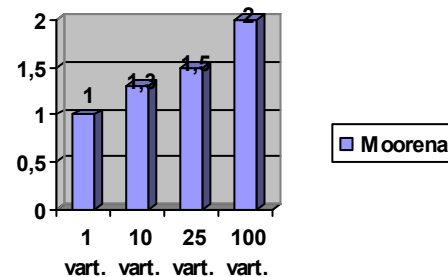
41 pav. Eksperimento d) vykdymo laikas (s)

5.2.2. SportsML eksperimentas

Dokumentas turi 67 elementus ir suskaidytas po 2-10 atitinkamai atributus. Dokumento informacija išskaidoma į dvi dalis: tekstinė ir skaitinė. Pirmojo eksperimento metu buvo imituotas prisijungusiųjų vartotojų skaičius ir stebimi gauti rezultatai.

Eksperimentas a) ir jo savybės:

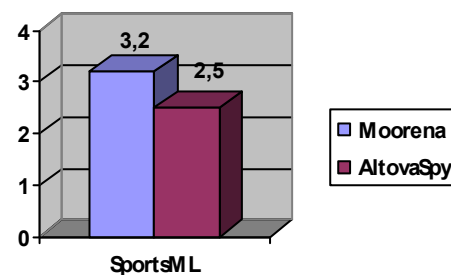
- 42 1, 10, 25, 100 imituoti vartotojų prisijungimai;
- 43 Interaktyvaus grafiko stebėjimas;
- 44 Tiesioginis varžybų rezultatai.



42 pav. Eksperimento a) vykdymo laikas (s)

Eksperimentas b) ir jo savybės:

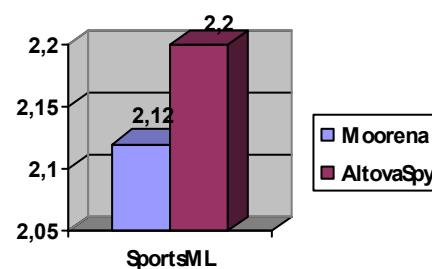
- Vienas vartotojas;
- Viso sezone žaistų rungtynių skaičius;
- Išrinkta 48 rungtynių rezultatai.
- Rezultatų patikrinimas.



43 pav. Eksperimento b) vykdymo laikas (s)

Eksperimentas c) ir jo savybės:

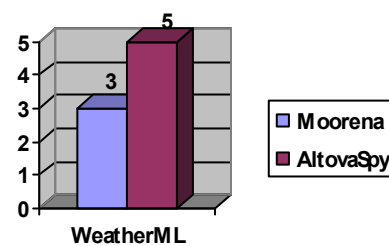
- Laiko intervalas (2009);
- Kaune žaistų rungtynių
- Išrinkti duoto laikotarpio temperatūros;
- Temperatūros svyravimai.
- Sudėtingumas: vidutinis



44 pav. Eksperimento c) vykdymo laikas (s)

Eksperimentas d) ir jo savybės:

- Duomenų nuskaitymas;
- 50 naujų rungtynių;
- Validacija ir įrašymas.



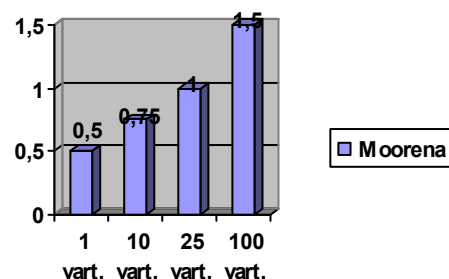
45 pav. Eksperimento c) vykdymo laikas (s)

5.2.3. Individualaus dokumento eksperimentas

Dokumentas turi 20 elementų ir po 7 kiekvienas atributus. Dokumento informacija išskaidoma į dvi dalis: tekstinė ir skaitinė. Pirmojo eksperimento metu buvo imituotas prisijungusių vartotojų skaičius ir stebimi gauti rezultatai.

Eksperimentas a) savybės:

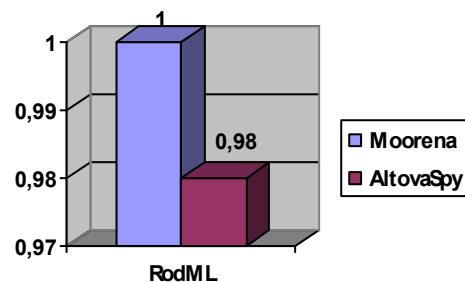
- 45 1, 10, 25, 100 imituoti vartotojų prisijungimai;
- 46 Interaktyvaus grafiko stebėjimas;
- 47 Skaitiklių parodymai.



46 pav. Eksperimento a) vykdymo laikas (s)

Eksperimentas b) savybės:

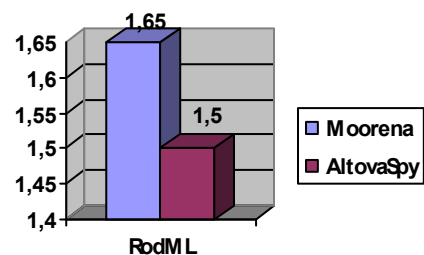
- Vienas vartotojas;
- Viršyta dienos norma;
- Rastas vienas įrašas.
- Rezultatų patikrinimas.



47 pav. Eksperimento b) vykdymo laikas (s)

Eksperimentas c) savybės:

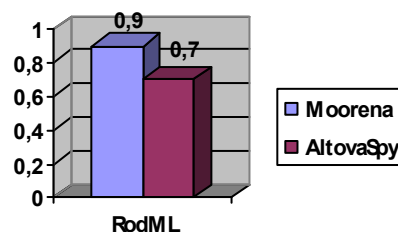
- Laiko intervalas (2009.05.05-2009.06.05);
- Taškas Z: stotelės numeris;
- Išrinkt tos stotelės parodymus;
- Temperatūra, slėgis, kiekisi.



48 pav. Eksperimento c) vykdymo laikas (s)

Eksperimentas d) savybės:

- Duomenų nuskaitymas;
- 10 dujų skaitiklių parodymai;
- Validacija ir įrašymas.



49 pav. Eksperimento c) vykdymo laikas (s)

5.3. Sugeneruotos ataskaitos

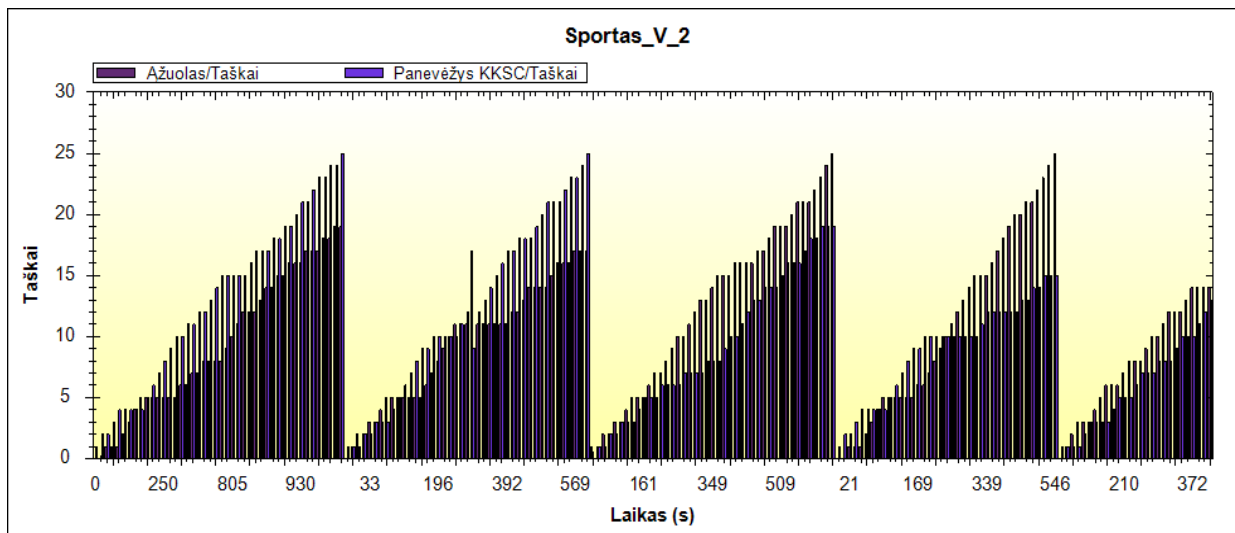
Šiame pavyzdyje pateikiama rungtynių statistika, komandų taškai, kėlinių suvestinės. Visi duomenys paimti iš XML standartizuotų (SportML) dokumentų.

Vyrų tinklinio čempionatas 2009 - 2010

20 lentelė.

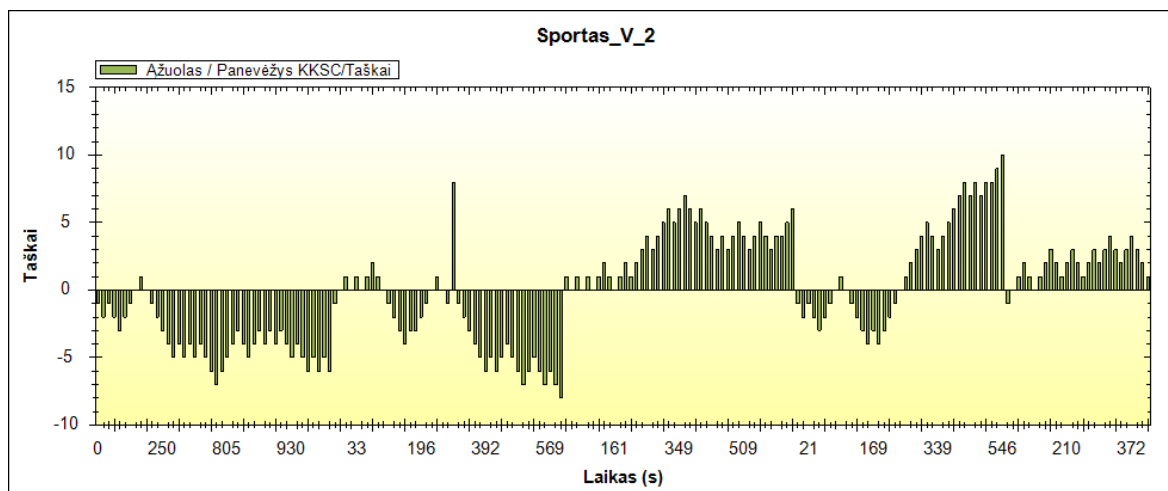
Data	Komanda	Taškai	Setas 1	Setas 2	Setas 3	Setas 4	Setas 5
2009.11.15	Ažuolas	101	19	17	25	25	15
2009.11.15	Panevezys KKSC	97	25	25	19	15	13

Pagal laiką (setus) pelnyti taškai:



50 pav. Taškų suvestinė

Komandų rezultatų skirtumas pagal rungtynių setus:



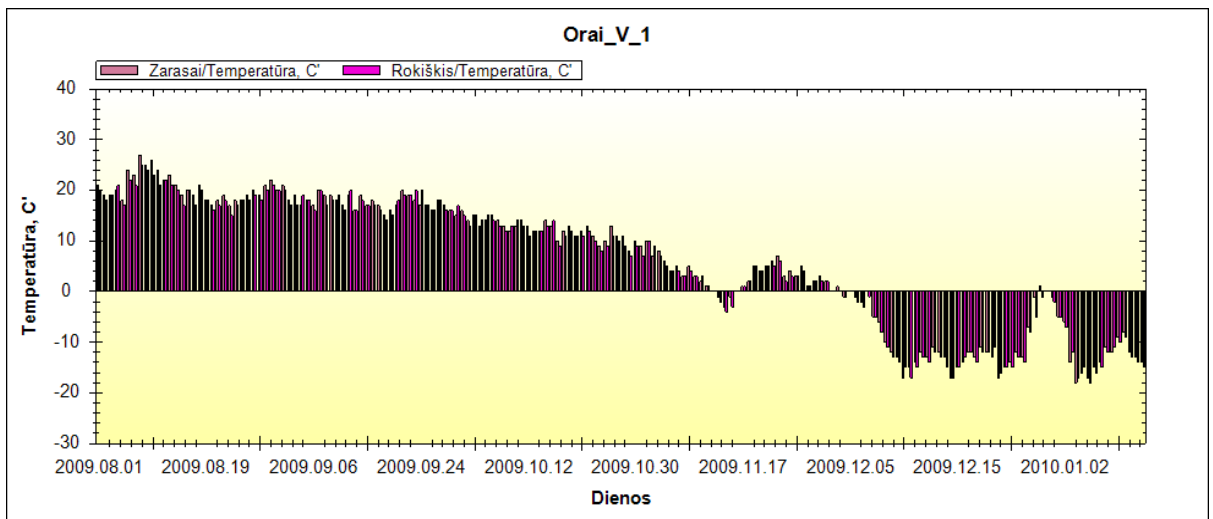
51 pav. Rezultatų skirtumas

Antrame pavyzdyje pateikiama maksimali, vidutinė ir minimali oro temperatūra. Visi duomenys paimti iš XML standartizuotų (WeatherML) dokumentų. Visa tai galima atsispausdinti PDF, Excel formatu pasirinktinai.

21 lentelė.

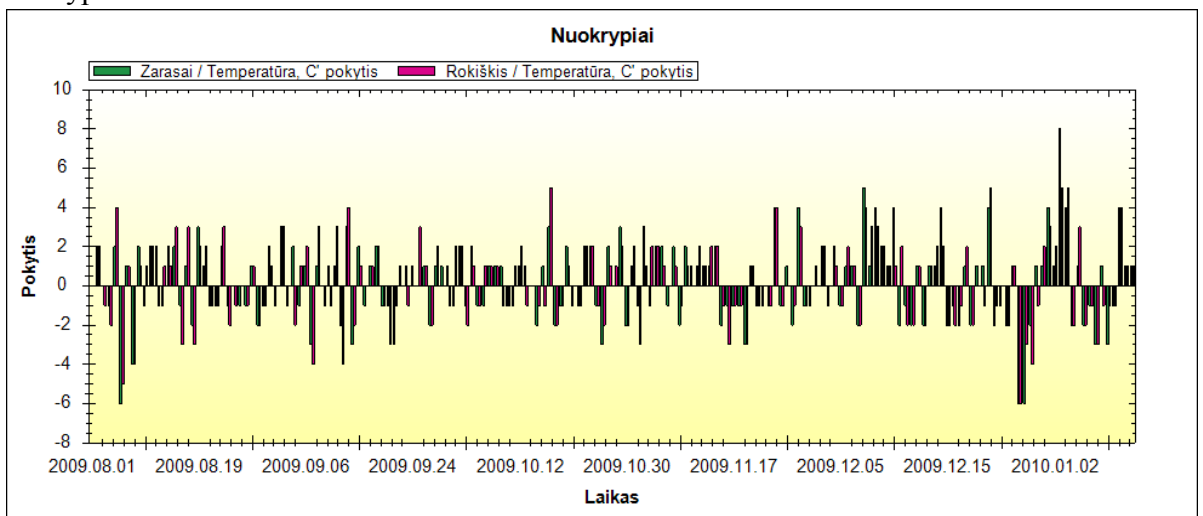
Šalis	Miestas	Max.	Min.	Vid.
Lietuva	Zarasai	Temperatūra, C' / 2009.08.08 / 27	Temperatūra, C' / 2010.01.04: -18	6,3807
Lietuva	Rokiškis	Temperatūra, C' / 2009.08.08 / 25	Temperatūra, C' / 2010.01.06: -18	5,8488

Temperatūrų svyravimai:



52 pav. Temperatūros svyravimai

Standartiniai nuokrypiai:



53 pav. Temperatūrų nuokrypiai

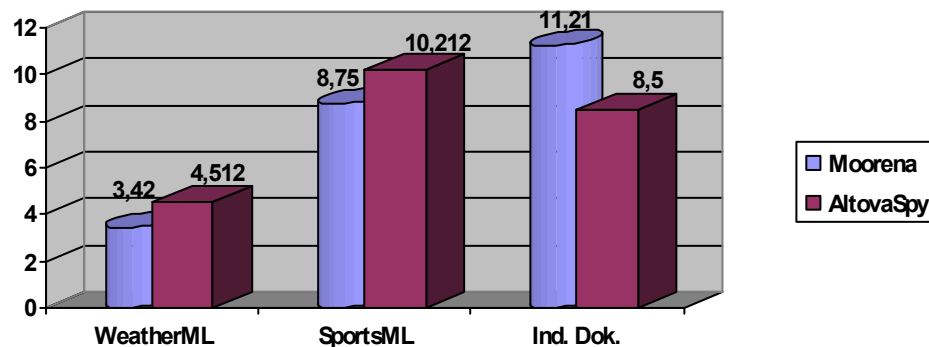
REZULTATŲ APIBENDRINIMAS IR DARBAI ATEIČIAI

Atlikus analizę ir tyrimus paaiškėjo, kad esamos programinės įrangos netenkina paprastų vartotojų keliamų reikalavimų, o kai kuriais atvejais ji yra netinkama. Egzistuojanti programinė įranga yra pritaikyta specializuotiems vartotojams, kurie dirba su XML dokumentais ir jų apdorojimu, tačiau nėra jokio sprendimo, kaip galima būtų bendradarbiauti, apsikeisti esama informacija ir rezultatais tarp paprastų vartotojų.

Toks autoriaus sukurtas įrankis leidžia kur kas greičiau apdoroti XML dokumentų duomenis, nei kiti sukurti įrankiai. PĮ leidžia juos laikyti duomenų saugykloje ir bet kuriuo metu išsitraukti norimus rezultatus. Vartotojai tarpusavyje gali bendradarbiauti ir naudotis komunikaciniais įrankiais: failų serveriu ir trumpųjų žinučių siuntimu, matyti dokumentų atnaujintą informaciją. Dėka interaktyvaus analizės įrankio grafiko, galima stebėti realiu laiku duomenis ir juos analizuoti, o prireikus automatizuoti t.y. sudaryti užduotis pagal vykdymo laiką ir vykdyti analizės arba duomenų migravimo operacijas.

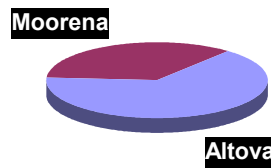
Eksperimento metu buvo pastebėta, kad PĮ charakteristikos žymiai prastesnės dirbant su didesnės apimties XML dokumentų duomenimis dėl naudojamos Microsoft framework technologijos užlaikymo verčiant programinį kodą į mašininį (Julia Lerman, 2008). Be to, programinė įranga turi apribojimą, kad negali atlikti analizės su XML dokumentais, kurie nėra standartizuoti, nes įrankis naudojami loginėmis struktūromis ir remiasi XSD, DTD schemomis. Tačiau ji žymiai patogesnė paprastam vartotojui. Kadangi PĮ yra atvirojo kodo, jai galima laisvai kurti papildomus analizės modulius – algoritmus ir taip praplėsti, bei vystyti sistemą.

Gauti bendri rezultatai patiesino lūkesčius, nes standartizuoti dokumentai ir programinė įranga geba greičiau apdoroti duomenis, be to toks programinės įrangos efektyvumas yra didesnis apsieičiant svarbia informacija su tarpininkais. Po atliktų eksperimentų aiškiai matomi, kad autoriaus programinė įranga „Moorena“ greitai geba analizuoti standartizuotus dokumentus greičiau, tačiau šiek tiek prasčiau susidoroja su nestandartizuotais dokumentais.



54 pav. Rezultatų suvestinė

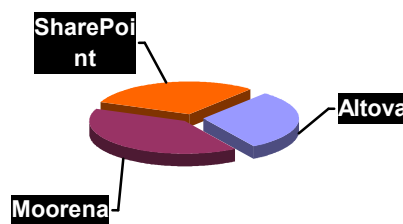
PĮ efektyvumo palyginimas su Altova analizės įrankiu:



55 pav. Efektyvumas

Paveiksle pateikiamas palyginimas analizuojant turimus XML dokumentus. Altova yra tikrai geras įrankis, kuris geba taip pat puikiai valdyti XML dokumento duomenis, kai kuriais atvejais net geriau, tačiau jis neturi aplinkos kuri leistų vartotojams bendradarbiauti tarpusavyje. Juk analizuojami duomenys yra naudojami ne vien tik savo reikmėms, o ir viešai skelbiami.

PĮ funkcionalumo palyginimas su kitais panašiais produktais:



56 pav. Funkcionalumo palyginimas

Autoriaus sukurta programinė įranga sujungia abiejų sistemų funkcionalumą ir programuotojų su analitikais darbą.

5.4. Ateities planai

Kiekvienas įrankis turi savo amžių ir su laiku ji atsinaujinta, taip tampa funkcionalesnė ir patogesnė. Kadangi autoriaus įrankis yra dar visai neseniai sukurtas, todėl jau dabar yra numatyti tolimesni jo praplėtimo ir tobulinimo darbai.

Pirma, programinė įranga sukurta ant .NET platformos skirtos Microsoft šeimos operacinėms sistemoms. Vartotojams gali iškilti problemų, norint pasinaudoti PĮ kitos šeimos OS (Mac, Linux). Kiekvienas šiuolaikinis žmogus visada turi šalia savęs delninį kompiuterį arba mobilųjį telefoną, kurie puikiai geba matyti ir valdyti interneto duomenų srautus. Tokia

PĮ būtų mobili ir įgautų daugiau gyvybės, jeigu būtų pasiekiami nepriklausomai nuo vartotojo turimos platformos t.y. panaudojant Silverlight/ASP.NET technologijas.

Antra, PĮ galima papildyti ir pritaikyti esamiems XML dokumentų standartams pildymo formas. Tokios PĮ pildymo formos leistų vartotojams ne tik analizuoti ir skaityti informaciją, bet ir pildyti ją rankiniu būdu arba prireikus duomenis taisyti patogesniu būdu (T.V. Raman, 2003). Dabartinėje PĮ galima taip pat keisti ir pildyti duomenis rankiniu būdu, tačiau reikia laikytis griežtos XML dokumento standarto struktūros (aprašyta 2 skyriuje), o duomenis pildyti teksto redaktoriumi. Be abejo, tai yra nepatogu, nes reikia skaityti XML dokumentų kodą ir išmanyti dokumento aprašymo taisykles. Tokių formų panaudojimas paslėptų šių dokumentų sudėtingas logines struktūras, nereiktų dėmesio skirti jų aprašymą ir leistų patogiai dirbti, įvedinėti reikiamus rezultatus.

Trečia, tyrime lygintoje Altova programinėje įrangoje buvo pastebėtas labai funkcionalus dalykas, tai XML dokumento struktūros grafiškas atvaizdavimas. Grafinis XML dokumento apiforminimas leidžia vartotojui geriau suprasti XML dokumento struktūrą ir jame esančius duomenis. Be to, toks funkcionalumas gali būti praplėstas į interaktyvų, kurio pagalba galima būtų susidėlioti reikalingus duomenis, o tai būtų lengviau ir aiškiau analizuoti dokumentų turinį ir išrinkti reikiamus rezultatus iš esamų duomenų.

6. IŠVADOS

Magistro darbo metu buvo ištirti statinio tipo dinaminiai dokumentai – XML. Analizuoti jų standartai, ieškomi privalumai ir trūkumai, esamų standartų palaikymas, bei nagrinėjamas tolimesnis duomenų panaudojimas pagal jų paskirtį. Atskleista duomenų laikinumo problematika. Tyrimo metu buvo pastebėta, kad nėra programinės įrangos, kuri leistų gerai valdyti XML dokumentų migravimą ir atlikti jų duomenų analizę, todėl buvo nuspręsta sukurti specializuotą analizės įrankį, kuris tenkintų apklausoje dalyvavusių vartotojų lūkesčius ir gebantis intuityviai ruošti ataskaitas. Eksperimento metu buvo palyginti rezultatai su panašios paskirties egzistuojančiais produktais.

Kuriant programinę įrangą, buvo paliesti ir ištirti iškilę šie klausimai ir problemos, bei duomenų apsikeitimo procesai:

- Supaprastintas XML dokumentų duomenų migravimas. Tai ypač aktualu vartotojams, kurie dirba su atvirais ir nepriklausomais dokumentais (federacijos, žiniasklaida, analitikai, meteorologai), kurie neturi nieko bendro su XML dokumentais apart jų duomenimis.
- Dirbantys vartotojai su PĮ pastebėjo, kad visas valdymo procesas yra paprastas ir patogus. Lengva bendradarbiavimo ir duomenų apsikeitimo aplinka. Leidžia vartotojams naudoti duomenis ne tik savo tikslams, tačiau dalintis jų rezultatais ir su kitais sistemos vartotojais.
- PĮ palaiko žinomus dokumentų standartus (SportsML, WeatherML, FinML, OrgML, NewsML, EventsML).
- Galima XML dokumentus transformuoti į kitus formatus. Palaikomi populiariausios ir žinomiausios ataskaitų ir pažymų formos pateikiamos PDF, Excel). Galima kurti Crystal Reports ataskaitų šablonus.
- Dėka interaktyvaus grafiko, galima stebėti rezultatus realiame laike. Taip pat juos analizuoti ir įvesti duomenis pagal pasirinktą formą.
- Programinė įranga automatizuotai ir atlieka importavimo, eksportavimo funkcijas. Sistema yra pakankamai savarankiška ir geba atlikti komandas intuityviai pagal sudarytą darbų grafiką, be kito vartotojo įsikišimo.
- Sukurta PĮ yra nemokama ir atvirojo kodo. Tai didelis privalumas, siekiant panaudoti programinę įrangą pritaikyti tolimesniems darbams ir ją plėsti.
- Sėkmingai realizuota programinė įranga pritaikyta ir įdiegta AB „Panevėžio energija“ ir AB „Lietuvos dujos“ pagal jų paskirtį.

LITERATŪRA

1. **Mitch, C. D'Cruz, E. Kay, M. D. Thomas**, XML: Problem – Design – Solution, *Wrox*, 2006, p. 280–295.
2. **Dr. K. Michael**, XPath 2.0 Programmers Reference, *Wrox*, 2004, p. 350-358.
3. **B. Marchal**, Applied XML Solutions, *SAMS*, 2000, p. 154-162.
4. **A. Freeman, A. Jones**, Microsoft .NET XML Web Services, 2005, p. 80-114.
5. **S. Holzner**, XPath (Navigating XML with XPath 1.0 and 2.0), *SAMS*, 2006, p. 95–108.
6. **M. C. Daconta, L. J. Obrst, K. T. Smith**, The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, *Wiley Publishing*, 2003, p. 18-25, 85-116.
7. **A. B. Chaudhri, A. Rashid, R. Zicari**, XML Data Management: Native XML and XML-Enabled Database Systems, *Wesley*, 2003, p. 416-430.
8. **W. R. Stanek**, XML Pocket Consultant, *Microsoft Press*, 2002, p. 277-281.
9. **C.F. Goldfarb**, XML Handbook (4th. Edition), *Congres Cataloging in Publication Data*, 2002, p. 464-487.
10. **L. M. Garshol**, Definitive XML Application Development, *Congres Cataloging in Publication Data*, 2001, p. 80-92, 564-583.
11. **E. R. Harold, W. S. Means**, XML in a Nutshell: A Desktop Quick Reference, *O'REILLY*, p. 26-35.
12. **K. Watson**, C# 2005 Databases, *Wax*, 2006, p. 45-52.
13. **Z. Bellahsene, E. Hunt, M. Rys, R. Unlend**, Database and XML Technologies: 6th International XML Database Symposium, *Spring*, 2009, p. 85-150.
14. **E. R. Harold**, Effective XML: 50 Specific Ways to Improve Your XML, *Adison Wesley*, 2004.
15. **R. Krištapaitis**, Moorena - software for Storing and Migrating Signed XML Documents, <http://moorena.roleka.lt/>, [Žiūrėta 2009-09-20].
16. **The Cover pages**, SportsML, <http://xml.coverpages.org/sportsML.html>, [Žiūrėta 2009-10-11].
17. Advanced Modeling and Browsing of Technical Documents, <http://www.pms.ifi.lmu.de/publikationen/PMS-FB/PMS-FB-2001-11.pdf>, [Žiūrėta 2009-11-04].
18. **Geoff Richards**, The big list of XML technologies, http://xmlsuk.org/xml_technologies/, [Žiūrėta 2009-11-10].

19. XML Security Specifications Maintenance Working Group,
<http://santuario.apache.org/who.html>, [Žiūrėta 2009-11-10].
20. The XML Security standards, <http://www.sitepoint.com/article/getting-started-xml-security/>, [Žiūrėta 2009-11-11].
21. XML Security Library, <http://www.aleksey.com/xmlsec/>, [Žiūrėta 2008-11-11].
22. **O'Reilly**, A technical intrudaction to XML,
<http://www.xml.com/pub/a/98/10/guide0.html?page=2>, [Žiūrėta 2009-09-08].
23. **Chris Hellmuth**(Colgate University),The Linguist's Toolbox and XML Technologies,
<http://www.linguistlist.org/emeld/workshop/2006/papers/hellmuth.html>,
[Žiūrėta 2009-10-05].
24. **J. Andrew Schafer**, <http://msdn.microsoft.com/en-us/magazine/cc302121.aspx>,
[Žiūrėta 2009-10-16].
25. **Niel Bornstein**, C# and XML,
<http://www.xml.com/pub/a/2002/03/06/csharpxml.html?page=1>, [Žiūrėta 2009-10-20].
26. Advanced Modeling and Browsing of Technical Documents,
<http://www.pms.ifi.lmu.de/publikationen/PMS-FB/PMS-FB-2001-11.pdf>, [Žiūrėta 2009-11-04].
27. **MSDN magazine**, XML in .NET,Framework XML classes,
<http://msdn.microsoft.com/en-us/magazine/cc302158.aspx>, [Žiūrėta 2009-05-14].
28. **IBM**, XML forums, http://www.ibm.com/developerworks/forums/dw_xforums.jspa,
[Žiūrėta 2009-04-03].
29. **D. Gulbransan**, Using XML, second edition, *QUE*, 2002, p. 8-14.
30. IPTC, *Internetal press telecommunications council*,
<http://www.iptc.org/cms/site/index.html?channel=CH0177>, [Žiūrėta 2010-01-10].
31. **Michael A. Cusumano**, The business of software, *FREE PRESS*, 2004, p. 39, 43, 46.

TERMINŲ IR SANTRUMPŲ ŽODYNAS

DB – reliacinė duomenų bazė.

XML – išplėstinė žymių kalba. Pagrindinė XML kalbos paskirtis yra užtikrinti lengvesnį duomenų keitimąsi tarp skirtingo tipo sistemų, dažniausiai sujungtų internetu.

PI – programinė įranga.

DBVS – reliacinės duomenų bazės valdymo sistema.

Play-by-Play – apibūdinimas detalaus veiksmo žaidimo progresu. Toks būdas paplitęs transliavime rungtynių, kai vaizdo įrašas nematomas, t.y. nėra galimybių žiūrėti tiesioginio vaizdo arba įrašo.

OS – operacinė sistema.

YAML – tai XML nauja atmaina, skiriasi savo specifikavimu.

CSS – (angl. *Cascading Style Sheets*) – kalba, skirta nusakyti kita struktūrine kalba aprašyto dokumento vaizdavimą.

NITF – tai XML žinių teksto dokumento standartas.

XSLT (ang. *eXtensible Stylesheet Language (XSL) transformations*) – kalba, aprašanti XML dokumento transformaciją į HTML dokumentą.

.NET Framework – Microsoft Windows operacinės sistemos komponentas. Jis suteikia kitoms programoms galimybę naudotis daugybe jau paruoštu įvairių bibliotekų (pvz., duomenų bazių komponentus, formų komponentus). Be to, šis komponentas ir tvarko programos kodą jos vykdymo metu, jei programa parašyta specialiai šiam paketui.

Task Scheduler – sudėtinė Microsoft Windows dalis, kuri suteikia galimybę planuoti paleisti programas ar skriptus iš anksto nustatytais laiko tarptais, kartais.

Service – tai bet kuri programa arba komponentė, kuri užkraunama kartu su OS dar prieš vartotojui įeinant į tinklą ar pamatant ekranine. Servisai užkraunami automatiškai ir neleidžia operatoriui kištis į užkrovimo procesą.

GUI – grafinė vartotojo sąsaja, kuri skirta vartotojui palengvinti dirbą su programine įranga, t.y. sąveikauti su jos elementais, bei komponentais. Tai šiuolaikinė technologija į kurią vis dažniau skiriama dėmesio ir kuri dažniausiai atspindi programos funkcionalumą ir paprastumą.

RSS – XML failų formatų šeima internetiniam duomenų rinkimui iš tinklaraščių.

DTD – paskirtis yra nustatyti iš kokių blokų gali būti sudarytas XML dokumentas.

SportML – tai sportinės informacijos XML dokumentas, specialiai paremtas duomenų apsikeitimo būdui.

PRIEDAS A – KONFERENCIJOS PRANEŠIMAS

R. Krištapaitis, Statinio tipo dinaminiai dokumentai ir jų analizės įrankio apžvalga: XML dokumentų panaudojimas ir pritaikymas pasirinktai sričiai, IVUS 2010, p. 210-216.

Statinio tipo dinaminiai dokumentai ir jų analizės įrankio apžvalga

XML dokumentų panaudojimas ir pritaikymas pasirinktai sričiai

Santrauka — Ši tezė apjungia teorinių ir praktinių žinių panaudojimą kuriant dinaminiais (XML) dokumentams statinės analizės įrankius. Sistema sąmoninga tuo, kad remiasi sukurta vartotojo logika (medžio struktūros schema), kuri yra apibrėžiama paties vartotojo ir geba valdyti dokumento duomenis, bei analizuoti jų turinį. Sistemos veikimo logika – mazgo pasirinkimo semantika, pasinaudojant sukurtais dokumentų standartais aprašytais XML kalba (DTD, XSD). Problema iškyla, kai kuriant dinaminių dokumentų analizės įrankius laikomasi griežtų taisyklių, tai reiškia negalima sulaužyti statinio tipo dokumento struktūros ir jo logikos. Problemos sprendimui yra panaudojama XPath technologija. Tokiu būdu yra sukuriamos statinio tipo užklauskos ir išrenkamas dokumentų turinys, kuris vėliau atvaizduojamas vartotojo ekrane. Pagrindinis darbas yra sukurti įrankį, kurio pagalba galima atlikti statinę dokumento analizę. Dokumento duomenimis galima migruoti į pasirinktą reliacinę duomenų bazę ar kitą pasirinktą (PDF, XSL) dokumento formatą. Tokie įrankiai - analizatoriai leidžia sutaupyti daug laiko apdorojant, išrenkant, išsaugant ir analizuojant turimus duomenis.

I. ĮŽANGA

Šis straipsnis motyvuojamas tuo, kad šiais laikais yra plačiai paplitę dokumentai pagrįsti XML technologija. Jų tiesioginė paskirtis yra naudojama informacijos apsiukeitimui. Taigi, dažniausiai išskylanti problema – reikalingi specifiniai įrankiai (moduliai, komponentai). Tikslas – ištirti dinaminių dokumentų (standartai: SportML, WeatherML, MarketsML ir kt.) saugojimą ir judėjimą, bei sukurti specifinį įrankį skirtą tokių dokumentų analizei, bei apdorojimui. Straipsnyje pristatomas įrankis, kurio pagalba dokumentus galima ne tik analizuoti, bet ir migruoti jų turinį iš vienos formos į kitas (pvz.: XML į DB, PDF, Excel). Įrankio paskirtis leisti vartotojui valdyti dokumento struktūrą, o jų duomenis panaudoti pakartotinai (reuse) tolimesniems darbams. Projekte koncentruojamasi ne tik į senų XML standartų palaikymą, bet ir naujų įtraukimą. Vadinasi, tai yra naujos programinės įrangos karkaso kūrimas, kuri sugebėtų valdyti XML dokumentus. Tokie įrankiai leistų vartotojams nepriklausomai nuo jų darbo srities (analitikams, žurnalistams, teisėjams arba kitos srities specialistams) ir patirties lengvai dalintis svarbia informacija, ją apdoroti, bei pateikti suprantamesne vartotojui forma. Sukurta programinė įranga leidžia vartotojui kurti ir kitos struktūros statinio tipo dokumentus, o vėliau sinchronizuoti jų duomenis su pasirinkta duomenų baze. Naudojantis šios temos metodologija, PĮ galima pritaikyti įvairioms sritims: orams, sportui, finansams, žiniasklaidai, technologijoms ir kitokio turinio duomenis saugoti ir analizuoti.

II. METODOLOGIJOS IR TECHNOLOGIJOS

Informacijos kiekis nuolatosis ir be perstojo auga. Viskas gerai, kai informacijos daug, tačiau dažnai susiduriame su problema, kai esame nuolatosis priversti ieškoti informacijos po

didžiulį jos srautą darbe, namuose, o kartais jaučiame nusivylimą, kai bandydami surasti operatyviai duomenis, dažnai jų nerandame arba tiesiog sugaištame labai daug laiko kol surandame reikalingą, teisingą informaciją. Tokia atsakomybė atitenka paieškos sistemoms, kurios vaidina pagrindinį vaidmenį šioje problemos sprendime. Problemos tikriausiai nebūtų, jeigu visa prieinama apsikeitimo informacija būtų griežtai standartizuota ir klasifikuota į tipus ir kategorijas. Norint visa tai įgyvendinti, reikia ne tiek daug – griežtų taisyklių, priežiūros ir palaikymo. Atsiranda natūralus poreikis klasifikuoti visą informaciją, o tai labai lengva įgyvendinti panaudojus XML technologijos pagrindus, kitaip tariant pritaikyti globalius standartizuotus informacijos apsikeitimo dokumentus. Gerai yra žinomi XML standartai, tokie kaip SportML (sportiniams įvykiams), WeatherML (orams), FinML (finansiniams), MarketsML (marketingas), EventsML (žiniasklaidos) yra būtent tie dokumentai, kurie yra skirti rinkti duomenis ir jais apsikeisti. Kaip bebūtų keista, reali situacija yra išties prasta, nes duomenų surinkimo ir pateikimo metodai skiriasi, o renkama informacija yra dažnai netiksli, dėl tos priežasties kartais prisibijoma ir vengiama naudoti sukurtus apsikeitimo dokumentus. Ižvelgiama duomenų laikinumo, pakartotinio panaudojimo ir migracijos problema. Šiame straipsnyje pamėginsime panagrinėti giliau šiuos dokumentus autoriaus sukurtu įrankiu, kuris padėtų ir praplėstų šių dinaminių dokumentų panaudojimą realiame gyvenime. Šie dokumentai paremti statine struktūra. Tuo pačiu paanalizuosime jau sukurtus prieš tai minėtus standartus ir išvelgti jų privalumus, bei trūkumus. Pamėginsime atsakyti ir panagrinėti atvirus klausimus: ar verta standartizuoti visą informaciją, kaip galima pritaikyti XML dokumentus pasirinktai sričiai, ar reikalingas toks įrankis, kuris gebėtų analizuoti dinaminių dokumentų (XML) turinį. Taip pat pabandysime pašalinti esamas problemas, kurios nuolat pasikartoja iškilus reikalui apsikeisti duomenimis. Taigi, bus pasiūlytas įrankis, kuris sutvarkys indeksuotus sąrašus (metaduomenys) taip, kad informacijos pralaidumas būtų optimizuotas. Indeksų susitarimas padės vartotojams greičiau susirasti reikalingus atsakymus ir gauti aiškius, palyginamus rezultatus.

a) Aktualijos

Tema yra aktuali visoms informacinėms sistemoms, kurių veiklą galima priskirti statistikos rinkimui, rezultatų kaupimui – spauda, informaciniai portalai, federacijos, organizacijos. Tokia programinė įranga, kaip pavyzdys galėtų būti panaudota varžybų metu stebint, registruojant ir analizuojant varžybų rezultatus. Registruojami varžybų rezultatai ir duomenys yra greitai ir operatyviai pasiekiami kitoms perspektyvioms informacinėms sistemoms – naujienų portalai, žiūrovai, treneriai, analitikai. Be to, toks principas nereikalauja papildomo informacijos apdorojimo darbo ir sistema gali būti pilnai automatizuota. Programos perspektyva labai didelė ir gali remtis naujais sukurtais XML standartais, tokiu būdu programą galima platinti varžybų organizatoriams, federacijoms, kurios oficialiai atsakingos už varžybų rezultatus, bei jų kaupiamus duomenis ir statistiką. Programinę įrangą galima tobulinti nesunkiai integravus vieną ar kitą modulį ir pritaikyti vis kitai sporto šakai ar sričiai, o tai jau didelė perspektyvą į plėtimąsi ir tolimesnį jos vystymą. Dar vienas šios programinės įrangos privalumas, kad vartotojas gali pats pasirašyti XML dokumentus, todėl nuolat besikeičiant reikalavimams galima nesunkiai pritaikyti vis kitiems poreikiams. Įrankis orientuojamas į paprastą vartotoją, kuris turi mažai žinių technologijose, todėl jos valdymas ir priežiūra nereikalauja jokių programavimo žinių ir pastangų, nes valdymas – lengvas ir automatizuotas.

b) Apžvalga

Daugelyje mano apžvelgtų straipsnių teko perskaityti įvairių nuomonių ir susidaryti bendrą vaizdą, kad tokia programinė įranga reikalinga. Paieškojus internete analogiškų

įrankių aptikti nepavyko, tačiau panašių produktų galima rasti (Altova, SpyXML, Simba). Pasiskaičius ir pabendravus atvirose specialistų forumuose [14] susidarė vaizdas, kad šie įrankiai yra universalūs ir labiau orientuoti patiems programuotojams, nei paprastiesiems vartotojams. Be to, įrankiai yra ne atviro kodo ir licenzijos mokamos, todėl tobulinti laisvai negalima. Tokios specializuotos programinės įrangos poreikis yra gana didelis, ypač nepelno siekiančiose organizacijose. Kyla klausimas natūralus kodėl? Galbūt todėl, kad XML standartai yra atviri ir duomenys laisvai dalijami, tačiau yra įvairių būdų kaip galima apsaugoti šiuos duomenis (šifruoti, koduoti), tačiau čia galima jau galvoti naujas tezes ir susidurti su kitomis problemomis, bei jų gveldenimo subtilybėmis.

c) Technologijos

Programinė įranga realizuota C# .NET, Framework 3.0 aplinkoje. Microsoft išleidęs Framework 3.0, 3.5 versijas, programuotojams pasiūlė išties nemažai pakeitimų. Vienas iš sprendimų pats svarbiausias mūsų realizuojamam projektui, tai naudingos klasės ir metodai, kurie leidžia dirbti su XML dokumentais ir jų duomenimis, o tai labai patogiu, nes nereikia diegti papildomų komponentų [13]. Microsoft išties šioje srityje stipriai padirbėjo. XML dokumentų duomenys sinchronizuojami į MS SQL Server, tačiau nebuvo atsisakyta apsiriboti viena reliacine duomenų baze, todėl buvo nuspręsta užklausas kurti naudojant LINQ technologijas, transformuoti užklausas iš Linq į SQL. Papildomai naudojama ir XPath, kuri leidžia valdyti patį dokumentą ir lipti arba leisti pakopomis dokumento struktūroje panašioje į medį. Visos šios technologijos yra plačiai paplitusios ir naudojamos daugelio programuotojų. Šios technologijos gana stipriai įsitvirtinusios programavimo rinkoje yra pačios populiariausios, kur apstu įvairių pasirinkimų sprendžiant vieną ar kitą klausimą, tai patogiausi sprendimai iš visų.

d) Vizija

Sėkmingai realizuotą programinę įrangą, jau ketinama perkelti į globalų tinklą – internetą. Vartotojams nereiks įsidiesti programinės įrangos savo kompiuteryje, o ir prieinamumas visiems bus daug patogesnis. Planuose numatyta išleisti pirmąją jos versiją jau šių metų pabaigoje. Realizuojama bus panaudojant Silverlight/ASP .NET technologijas. Be to, vartotojui bus pasiūlytas pasirinkimas duomenų sinchronizavimas su kitais serveriais (MySQL, Oracle).

III. XML DOKUMENTŲ STRUKTŪRA

XML dokumentų struktūra apibrėžiama loginėmis schemomis (DTD, XSD). Kad būtų lengviau ir paprasčiau suvokti programinės įrangos veikimo principą, pradžioje susipažinkime su sukurtu panašiu XML dokumento standartu. SportML, tai standartas sukurtas globaliam naudojimui, apsikeitimui varžybų (krepšinio, futbolo, teniso, rankinio) rezultatais.

a) Statinių dokumentų kūrimas

Pamėginsime panagrinėti XML dokumentų struktūrą arba kaip turėtų būti sudaryti statinio tipo dokumentai ir jų schemas. Pavyzdžiui: pasiimkime jau sukurtą SportML standartą, kuris atspindi gerą loginę dokumento struktūrą ir pagal ją galima sudaryti kitus standartus ir dokumentų struktūras.

SportsML naudoja bendros paskirties duomenų struktūrą, bei jo turinio aprašomąją kalbą (XML), kad galėtų apibrėžti turinį ir sportinių duomenų struktūrą, tai reiškia, kad

interaktyvių ar sportinių rezultatų paslaugų supirkėjai turi daug lengvesnį darbą, sujungdami gautus rezultatus. SportsML susideda iš pagrindinio DTD dokumento (statinis tipas), turinčio savyje labai daug savybių, kurios apibūdina platų sportinės žiniasklaidos diapazoną.

Kalbant visapusiškai apie XML žodynų sudarymo privalumus, galima teigti, kad kol kas yra maža dalis pilnai aprašytų standartų, tačiau niekas neriboja šių standartų kūrimo ir praplétimo. Pati pradžia jau padaryta, todėl analogiškai galima įtraukti ir kitas sritis. Kadangi standartai yra atviri ir suinteresuotų žmonių vis daugėja ir auga, bei plečiasi, vystymas žengia į vis tolyn [2].

Pagrindinės statinio tipo dokumentų savybės:

- Globalios apimties planas
- Bendra struktūra visai pasirinktai sričiai
- Papildomai kuriami moduliai
- Gerai dokumentuotas ir lengva panaudoti
- Atviras ir nepriklausantis
- Pastoviai palaikomas ir plečiamas

Dabar pažiūrėkime ir panagrinėkime pavyzdį (1 pav.), kuris atspindi pagrindinę SportML dokumentų architektūrą, tačiau tai yra tik mažas fragmentas viso standarto iš to ką galima atvaizduoti panaudojus detalesnį aprašymą. Tai yra logiškas duomenų atvaizdavimas. Be to, tai nėra svarbu, kaip tie duomenys yra apipavidalinti. Pažvelkime, kaip informacinė sistema gali panaudoti dokumentą, kad atvaizduotų, kaip dvejų komandų susitikimo varžybas [5], tada naudojami dokumento keli elementai (sportinis įvykis < sports-event > ir komanda < team >):

```
<sports-event>
<sports-metadata event-status="post-event"/>
<team>
<team-metadata>
<name first="Kauno" last="Žalgiris"/>
</team-metadata>
<team-stats score="84" event-outcome="win"/>
</team>
<team>
<team-metadata>
<name first="Prancūzija" last="SLUC Basket"/>
</team-metadata>
<team-stats score="81" event-outcome="loss"/>
</team>
</sports-event>
```

Paveikslėlis 1. SportsML fragmentas

Šis SportsML fragmentas išverstas į HTML kalbą atvaizduojamas pirmoje lentelėje:

I lentelė. SportML į HTML

Kauno Žalgiris	84
Prancūzija SLUC Basket	81

Įvykio būsenos atributas „post-event“ panaudotas fragmente yra reikalingas tam, kad galutiniame rezultate gautūsi perverstas rezultatas iš SportML į HTML dokumento formatą [11].

Pagrindinis struktūrinis ir karkasinis elementas naudojamas <sports-content>, kuris reikalauja sekcijos <sports-metadata> ir kitų elementų:

- <sports-event>
- <tournament>
- <schedule>
- <standing>
- <statistic>
- <article>

Pirmi penki paminėti elementai sudaro XML struktūrą, kuri įvairiomis kombinacijomis apibūdina komandų <team> ar žaidėjų <player> rezultatus. Paskutinis elementas <article> daugiau naudojamas straipsnių rašyme (NITF).

Duomenų struktūra yra apibrėžta šiais bruožais:

II lentelė. Duomenų struktūra

<sports-event>	Žaidimo komandos arba žaidėjų informacija apie vietas, pasirenkamos informacijos apie pareigūnus siejasi, žaidimas realiu laiku, pagrindiniai momentai ir apdovanojimai.
<tournament>	Turnyro rezultatai, kurie žaidžiami per kelis kartus, raundus.
<schedule>	Suformuotas sportinių įvykių komplektas (kada, kur).
<standing>	Žaidėjų arba komandos pozicija.
<statistic>	Žaidėjų arba komandos statistika.

b) Statinių dokumentų kūrimo problematika

Kiekvienas dokumentas turi savo metaduomenis, todėl atitinkamai sudaromi ir atskiri statiniai dokumentai, kurie pritaikomi pagal pasirinktas taikymo sritis (mokslas, statistika, prognozės). Dokumentai gali turėti skirtingą turinį, todėl skiriasi ir jų logines schemas. Tokiu atveju iškyla dažna problema – dokumentų apdorojimas, jų analizė ir panaudojimas tolimesniems darbams. Šios problemos sprendimui galima panaudoti specializuotą programinę įrangą, kuri išsprendžia visas šias minėtas problemas ir padeda vartotojui pateikti reikiamus rezultatus.

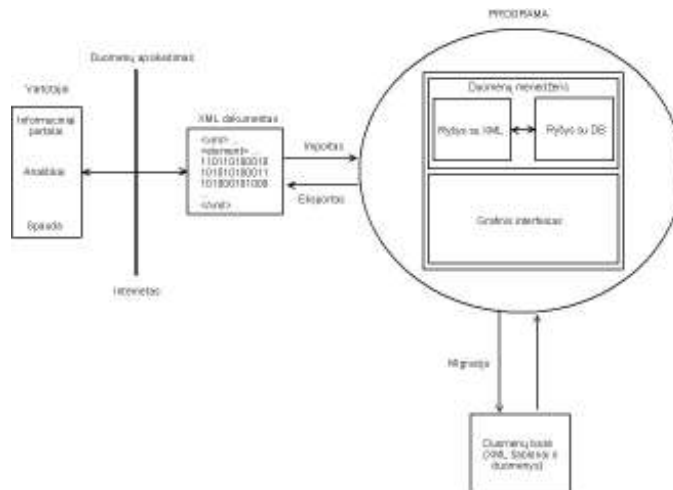
IV. ANALIZĖS ĮRANKIS

Svarbiausia yra išsiaiškinti kokie dažniausiai reikalavimai, kriterijai lemia šių dokumentų tolimesnį panaudojimą. Pirma, tai dokumentų turinio analizė arba kitaip tariant reikalingų rezultatų pateikimas. Antra, duomenų saugojimas ir apsikeitimas.

Svarbiausios įrankio savybės:

- Importas
- Eksportas
- Analizė (grafikai, algoritmai)
- Automatizavimas
- Centralizavimas

Visos paminėtos savybės tarpusavyje yra glaudžiai susiję, nes norint analizuoti duomenis, pirma reikia juos pasiimti nuo kamieno ir pereiti prie šakų pagal loginę dokumento struktūrą. Tada galima išsirinkti reikiamą informaciją ir pateikti tinkama forma (grafikai, tekstinis pavidalas), bei išsaugoti (PDF, Excel, XML, RDB) reikiamu formatu.

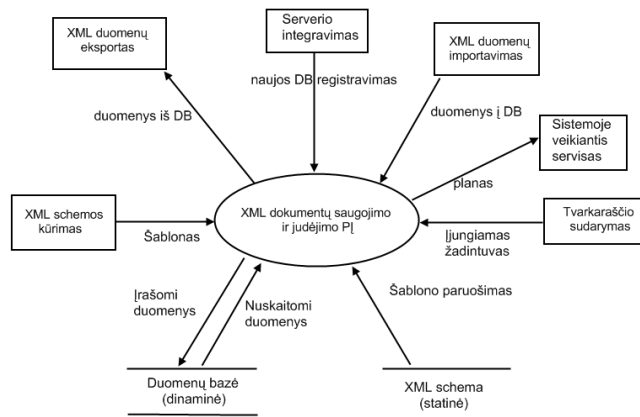


Paveikslėlis 2. Įrankio panaudojimo logika

Šiame paveikslėlyje (2 pav.) pavaizduotas įrankio veikimo procesas. Šioje sistemoje dalyvauja 4 resursai: vartotojas, duomenų šaltinis, programinė įranga ir duomenų bazė. Toliau susipažinkime su detalesniu įrankio aprašymu.

a) Kontekstas

Kontekstas atspindi programinės įrangos veiklos sferą. Žemiau (III lentelė) pateikiamas detalesnis veiklos aprašas. Pateikiamas veiklos sąrašas ir aprašyti srautai.



Paveikslėlis 3. Konteksto diagrama

III lentelė. Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys / išeinantys informacijos srautai
1.	Serverio integravimas	Serverio duomenys (int)
2.	Sukuriamas naujas XML dokumentas	Įrašomi atributai į duomenų bazę (in)
3.	Duomenų importavimas	Informacija įrašoma į DB (in)

Eil. Nr.	Įvykio pavadinimas	Įeinantys / išeinantys informacijos srautai
4.	Duomenų eksportavimas	Informacija įrašoma į dokumentą (out)
5.	Tvarkaraščio integravimas (nustatyta data ir laikas užduoties įvykdymui)	Išsaugojami atributai į DB (in)
6.	Serviso paleidimas	Paruoštas darbų planas, tvarkaraštis (out)

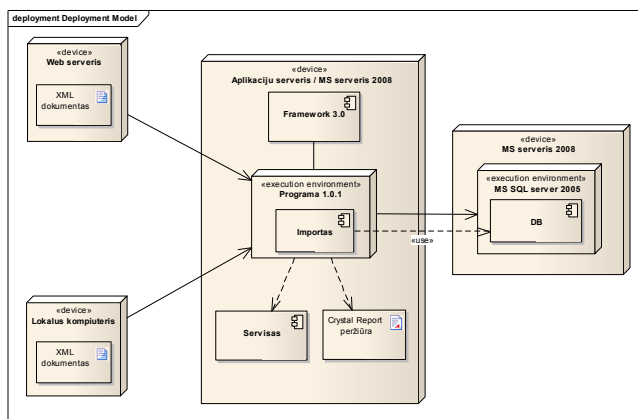
b) Įrankio bendras išdėstymo vaizdas

Komponentus ir techninę įrangą galima paskirstyti į keturias grupes. Pirmasis komponentas yra WEB serveris – nutolęs kompiuteris, kuris yra pasiekiamas per tinklą per kurį galima pasiimti XML dokumento duomenis. WEB serverį galima pakeisti ir lokaliu kompiuteriu, tai yra vietiniame tinkle esančiu kompiuteriu, kuriame yra įrašytas XML dokumentas arba vartotojas. Siekdami didesnio patikimumo ir saugomo lygio galima naudoti du atskirtus serverius. Vienas skirtas aplikacijai, o kitas duomenų bazei saugoti. Aplikacijų serveryje instaliuota programinė įranga skirta valdyti duomenims, tai yra migracijai ir priskirti naujas užduotis OS servisui, tau yra automatizuoti (angl. sheduler) programinę įrangą, be vartotojo įsikišimo. Tačiau aišku, tai tik yra siūloma schema, kaip padaryti patikimą globalią sistemą. Galima tokio turinio programinę įrangą naudoti asmeniniams tikslams ir diegti PĮ į lokalų kompiuterį.

Techninis architektūros sprendimas:

- Windows server 2005-2008
- Microsoft .NET technologijos
 - Visual Studio .NET
 - SQL Data provider
 - XML data reader, writer
- Crystal Report for Visual Studio .NET

Šios technologijos paveikia tiek paviršutinius, tiek ir vidinius programinės įrangos sistemos sluoksnius. Tai atsispindi grafiniame paveikslėlyje, kuriame matome kaip kuri sistemos dalis sąveikauja ir yra atsakinga už savo dalies aprūpinimą.



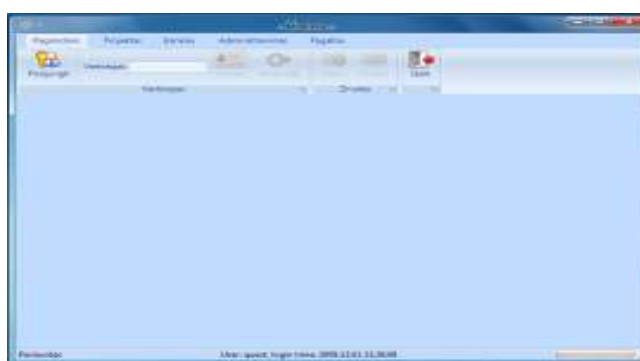
Paveikslas 4. Komponentų ir techninės įrangos sluoksniai

Bandomąją (beta) šios programinės įrangos versiją galima atsisiųsti autoriaus puslapyje (<http://moorena.roleka.lt>). Puslapyje pateikiama kita smulki informacija susijusi su produktu, tai versijų atnaujinimai, klaidos ir funkcijos.



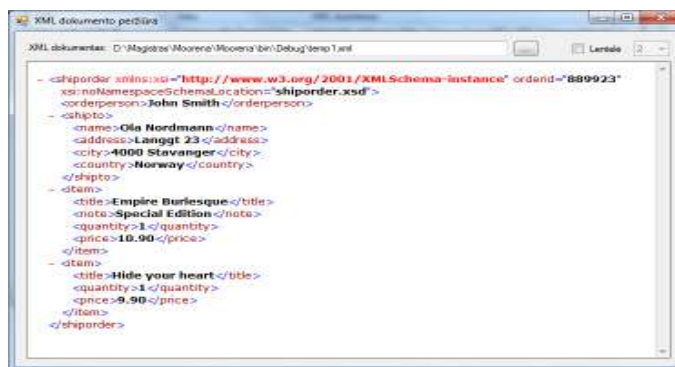
Paveikslas 5. Bandomosios PĮ puslapis

Programinės įrangos vartotojo sąsaja yra funkcionali ir įprasta (Office 2007 išvaizda), tokia išvaizda draugiška ir priimtina vartotojo atžvilgiu.



Paveikslas 6. Vartotojo sąsaja

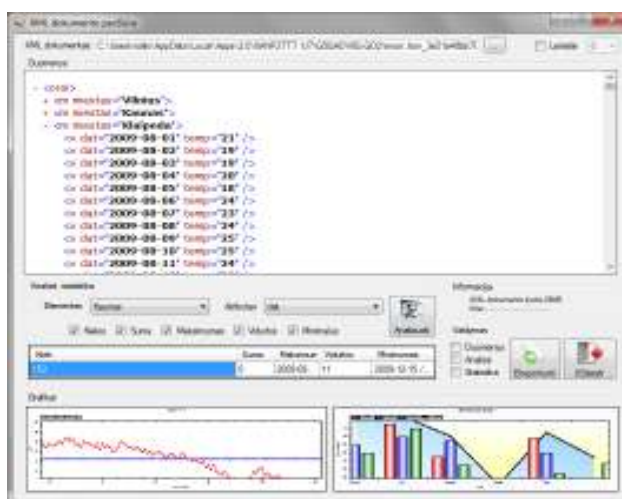
Įrankis geba atlikti įvairių dokumentų priežiūrą ir analizę. Galima taisyti jo turinį, logines schemas arba išsirinkti tam tikrus duomenis pagal pasirinktą statinio dokumento schemą. Įrankis prisitaiko prie kiekvieno dokumento standarto ir geba atlikti duomenų analizę ir statistiką.



Paveikslas 7. Dinaminių dokumentų turinio peržiūra

Šiuo įrankiu galima ne tik atlikti pagal įvairius sluoksnius dokumento analizę, bet ir išsaugoti jo turinį į pasirinktą formatą (XML, PDF, Excel). Duomenys saugomi realizacinėje duomenų bazėje ir bet kuriuo metu sistema automatizuotai gali formuoti tam tikras užduotis (pasirenka vartotojas):

- Importas
- Eksportas



Paveikslas 8. PĮ analizės įrankis

Literatūros šaltiniai

- [1] Rolandas Krištapaitis, Moorena – software for Storing and Migrating Signed XML Documents, <http://moorena.roleka.lt/>, [Žiūrėta 2009-09-20].
- [2] The Cover pages, SportsML, <http://xml.coverpages.org/sportsML.html>, [Žiūrėta 2009-10-11].
- [3] Advanced Modeling and Browsing of Technical Documents, <http://www.pms.ifi.lmu.de/publikationen/PMS-FB/PMS-FB-2001-11.pdf>, [Žiūrėta 2009-11-04].
- [4] Geoff Richards, The big list of XML technologies, http://xmlsuk.org/xml_technologies/, [Žiūrėta 2009-11-10].
- [5] XML Security Specifications Maintenance Working Group <http://santuario.apache.org/who.html>, [Žiūrėta 2009-11-10].
- [6] The XML Security standards, <http://www.sitepoint.com/article/getting-started-xml-security/>, [Žiūrėta 2009-11-11].
- [7] XML Security Library, <http://www.aleksey.com/xmlsec/>, [Žiūrėta 2008-11-11].

- [8] O'Reilly , A technical intrudaction to XML,
<http://www.xml.com/pub/a/98/10/guide0.html?page=2>, [Žiūrėta 2009-09-08].
- [9] The Linguist's Toolbox and XML Technologies, Chris Hellmuth (Colgate University),
<http://www.linguistlist.org/emeld/workshop/2006/papers/hellmuth.html>,
[Žiūrėta 2009-10-05].
- [10] J. Andrew Schafer, <http://msdn.microsoft.com/en-us/magazine/cc302121.aspx>,
[Žiūrėta 2009-10-16].
- [11] Niel Bornstein, C# and XML,
<http://www.xml.com/pub/a/2002/03/06/csharpxml.html?page=1>, [Žiūrėta 2009-10-20].
- [12] Advanced Modeling and Browsing of Technical Documents,
<http://www.pms.ifi.lmu.de/publikationen/PMS-FB/PMS-FB-2001-11.pdf>, [Žiūrėta 2009-11-04].
- [13] XML in .NET,Framework XML classes,MSDN magazine,
<http://msdn.microsoft.com/en-us/magazine/cc302158.aspx>, [Žiūrėta 2009-05-14].
- [14] XML forums, IBM, http://www.ibm.com/developerworks/forums/dw_xforums.jspa,
[Žiūrėta 2009-04-03].

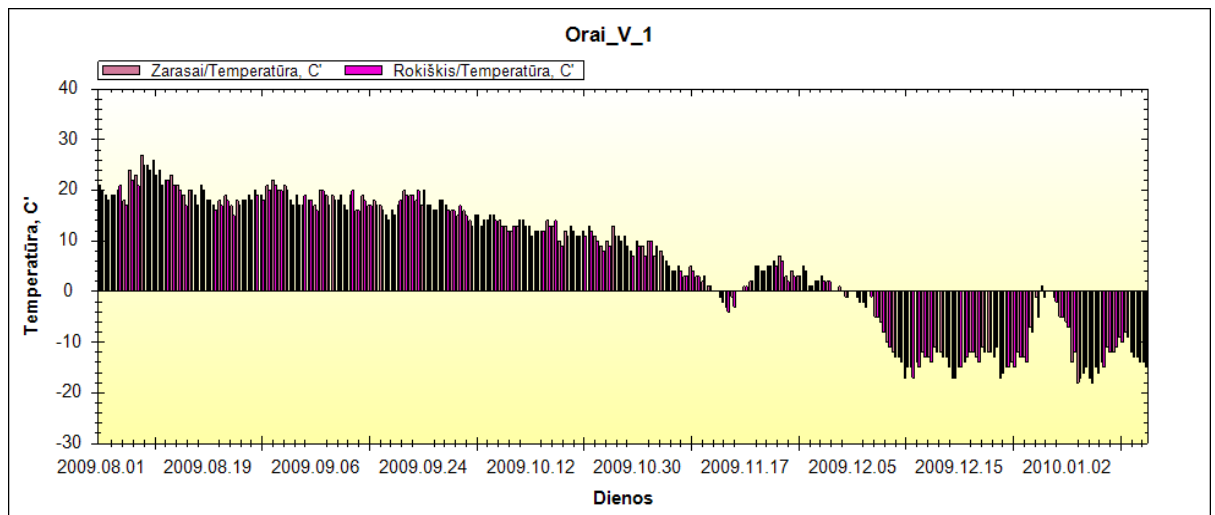
Norėčiau pasinaudoti proga ir padėkoti žmonėms, kurie prisidėjo tiesiogiai prie šio projekto vystymo. Esu dėkingas doc. dr. Eimučiui Karčiauskui už malonų bendradarbiavimą, kuriant pasirinktą analizės įrankį už didelį rūpestį ir dėmesį, taikant šios tezės praktinį panaudojimą sporto šakose. Taip pat noriu padėkoti Kęstučiui Motiejūnui ir Virginijai Limanauskienei už pastabas ir pataisymus.

PRIEDAS B – XML DOKUMENTŲ ANALIZĖS REZULTATAI

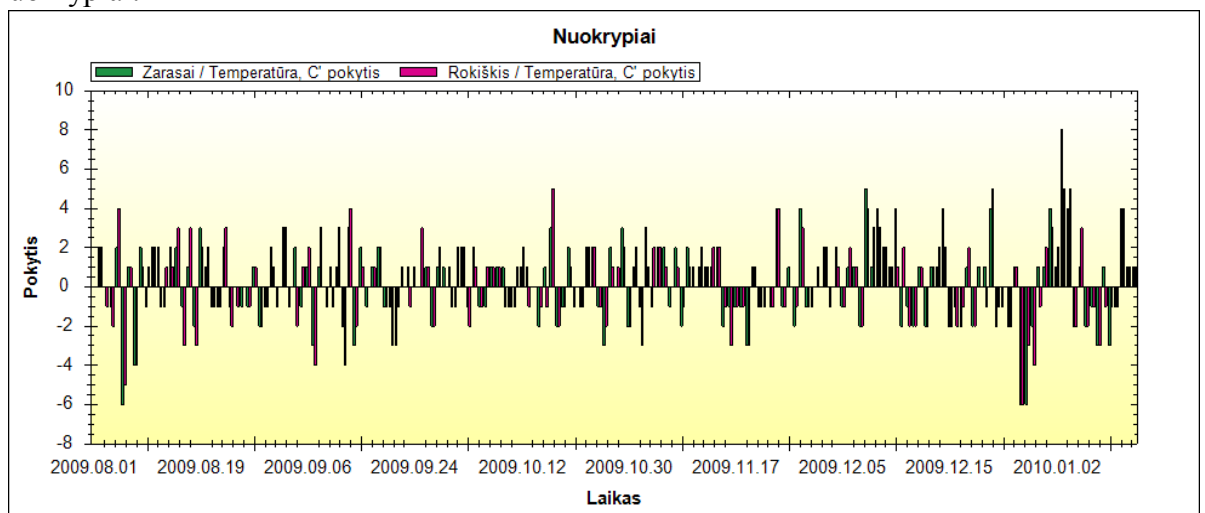
Šiame priede pateikiami išsamesni dokumento analizės rezultatai. Tai yra tik keli skirtingi pavyzdžiai, kaip galima sluoksniuoti ir atvaizduoti dinaminių dokumentų (XML) duomenis su autoriaus sukurtu įrankiu. Pirmame pavyzdyje pateikiama maksimali, vidutinė ir minimali temperatūra. Visi duomenys paimti iš XML standartizuotų (WeatherML) dokumentų. Visa tai galima atsispausdinti PDF, Excel formatu pasirinktinai.

Šalis	Miestas	Max.	Min.	Vid.
Lietuva	Zarasai	Temperatūra, C' / 2009.08.08 / 27	Temperatūra, C' / 2010.01.04: - 18	6,3807
Lietuva	Rokiškis	Temperatūra, C' / 2009.08.08 / 25	Temperatūra, C' / 2010.01.06: - 18	5,8488

Temperatūriniai
svyravimai:



Standartiniai
nuokrypiai:



Šioje lentelėje pateikiama dalis išrinktų iš XML (WeatherML) dokumento rezultatų:

city	country	dat	temp	wind	press	visa	prec
Rokiškis	Lithuania	2009.08.01	20	2	765	2000	0
Rokiškis	Lithuania	2009.08.02	18	3	756	1500	2
Rokiškis	Lithuania	2009.08.03	19	4	896	1700	1
Rokiškis	Lithuania	2009.08.04	21	3	859	1800	0
Rokiškis	Lithuania	2009.08.05	17	2	789	1500	10
Rokiškis	Lithuania	2009.08.06	22	5	685	1000	20
Rokiškis	Lithuania	2009.08.07	21	0	991	2000	0
Rokiškis	Lithuania	2009.08.08	25	1	871	2000	0
Rokiškis	Lithuania	2009.08.09	24	0	897	2000	0
Rokiškis	Lithuania	2009.08.10	23	3	759	1800	5
Rokiškis	Lithuania	2009.08.11	21	5	698	1600	4
Rokiškis	Lithuania	2009.08.12	22	6	756	1400	10
Rokiškis	Lithuania	2009.08.13	21	9	851	2000	0
Rokiškis	Lithuania	2009.08.14	20	5	862	2000	0
Rokiškis	Lithuania	2009.08.15	17	3	785	2000	0
Rokiškis	Lithuania	2009.08.16	20	2	955	2000	0
Rokiškis	Lithuania	2009.08.17	17	0	984	2000	0
Rokiškis	Lithuania	2009.08.18	20	0	752	2000	0
Rokiškis	Lithuania	2009.08.19	18	1	648	2000	12
Rokiškis	Lithuania	2009.08.20	16	3	785	600	20
Rokiškis	Lithuania	2009.08.21	17	3	854	2000	0
Rokiškis	Lithuania	2009.08.22	18	5	785	2000	0
Rokiškis	Lithuania	2009.08.23	15	2	782	2000	0
Rokiškis	Lithuania	2009.08.24	17	1	754	2000	2
Rokiškis	Lithuania	2009.08.25	18	0	758	2000	0
Rokiškis	Lithuania	2009.08.26	18	0	857	2000	5
Rokiškis	Lithuania	2009.08.27	19	3	925	2000	10
Rokiškis	Lithuania	2009.08.28	18	5	915	2000	0
Rokiškis	Lithuania	2009.08.29	20	6	942	2000	0
Rokiškis	Lithuania	2009.08.30	21	4	812	2000	0
Rokiškis	Lithuania	2009.08.31	20	2	842	2000	0
Rokiškis	Lithuania	2009.09.01	20	2	991	2000	2
Rokiškis	Lithuania	2009.09.02	17	1	991	2000	1
Rokiškis	Lithuania	2009.09.03	17	2	991	2000	15
Rokiškis	Lithuania	2009.09.04	19	6	991	2000	2
Rokiškis	Lithuania	2009.09.05	18	4	991	2000	0
Rokiškis	Lithuania	2009.09.06	16	8	991	2000	0
Rokiškis	Lithuania	2009.09.07	20	6	991	2000	0
Rokiškis	Lithuania	2009.09.08	17	1	991	2000	0
Rokiškis	Lithuania	2009.09.09	18	0	991	2000	0
Rokiškis	Lithuania	2009.09.10	19	0	991	2000	2
Rokiškis	Lithuania	2009.09.11	16	6	991	2000	5
Rokiškis	Lithuania	2009.09.12	20	7	991	2000	0

Rokiškis	Lithuania	2009.09.13	16	9	991	2000	0
Rokiškis	Lithuania	2009.12.17	-13	0	991	2000	40
Rokiškis	Lithuania	2009.12.18	-14	2	991	2000	40
Rokiškis	Lithuania	2009.12.19	-12	1	991	2000	55
Rokiškis	Lithuania	2009.12.20	-13	3	991	2000	59
Rokiškis	Lithuania	2009.12.13	-15	2	991	2000	72
Rokiškis	Lithuania	2009.12.14	-17	3	991	2000	72
Rokiškis	Lithuania	2009.12.15	-15	3	991	2000	72
Rokiškis	Lithuania	2009.12.16	-13	1	991	2000	80
Rokiškis	Lithuania	2009.12.17	-12	4	991	2000	80
Rokiškis	Lithuania	2009.12.18	-14	5	991	2000	80
Rokiškis	Lithuania	2009.12.19	-12	3	991	2000	95
Rokiškis	Lithuania	2009.12.20	-12	2	991	2000	95
Rokiškis	Lithuania	2009.12.21	-11	2	991	2000	95
Rokiškis	Lithuania	2009.12.22	-16	2	991	2000	99
Rokiškis	Lithuania	2009.12.23	-15	2	991	2000	99
Rokiškis	Lithuania	2009.12.24	-15	2	991	2000	100
Rokiškis	Lithuania	2009.12.25	-13	2	991	2000	100
Rokiškis	Lithuania	2009.12.26	-14	2	991	2000	100
Rokiškis	Lithuania	2009.12.27	-8	2	991	2000	120
Rokiškis	Lithuania	2009.12.28	-5	2	991	2000	120
Rokiškis	Lithuania	2009.12.29	-1	2	991	2000	80
Rokiškis	Lithuania	2009.12.30	0	2	991	2000	78
Rokiškis	Lithuania	2009.12.31	-2	2	991	2000	78
Rokiškis	Lithuania	2010.01.01	-5	1	991	2000	78
Rokiškis	Lithuania	2010.01.02	-7	2	991	2000	151
Rokiškis	Lithuania	2010.01.03	-12	3	991	2000	207
Rokiškis	Lithuania	2010.01.04	-17	2	991	2000	298
Rokiškis	Lithuania	2010.01.05	-15	1	991	2000	404
Rokiškis	Lithuania	2010.01.06	-18	0	991	2000	450
Rokiškis	Lithuania	2010.01.07	-16	2	991	2000	500
Rokiškis	Lithuania	2010.01.08	-15	0	991	2000	600
Rokiškis	Lithuania	2010.01.09	-12	1	991	2000	758
Rokiškis	Lithuania	2010.01.10	-11	2	991	2000	758
Rokiškis	Lithuania	2010.01.11	-10	3	991	2000	765
Rokiškis	Lithuania	2010.01.12	-9	2	991	2000	785
Rokiškis	Lithuania	2010.01.13	-13	2	991	2000	785
Rokiškis	Lithuania	2010.01.14	-14	3	991	2000	785
Rokiškis	Lithuania	2010.01.15	-15	2	991	2000	785
Zarasai	Lithuania	2009.08.01	21	2	765	2000	0
Zarasai	Lithuania	2009.08.02	19	3	756	1500	2
Zarasai	Lithuania	2009.08.03	19	4	896	1700	1
Zarasai	Lithuania	2009.08.04	20	3	859	1800	0
Zarasai	Lithuania	2009.08.05	18	2	789	2000	12
Zarasai	Lithuania	2009.08.06	24	5	685	2000	14
Zarasai	Lithuania	2009.08.07	23	0	991	2000	0

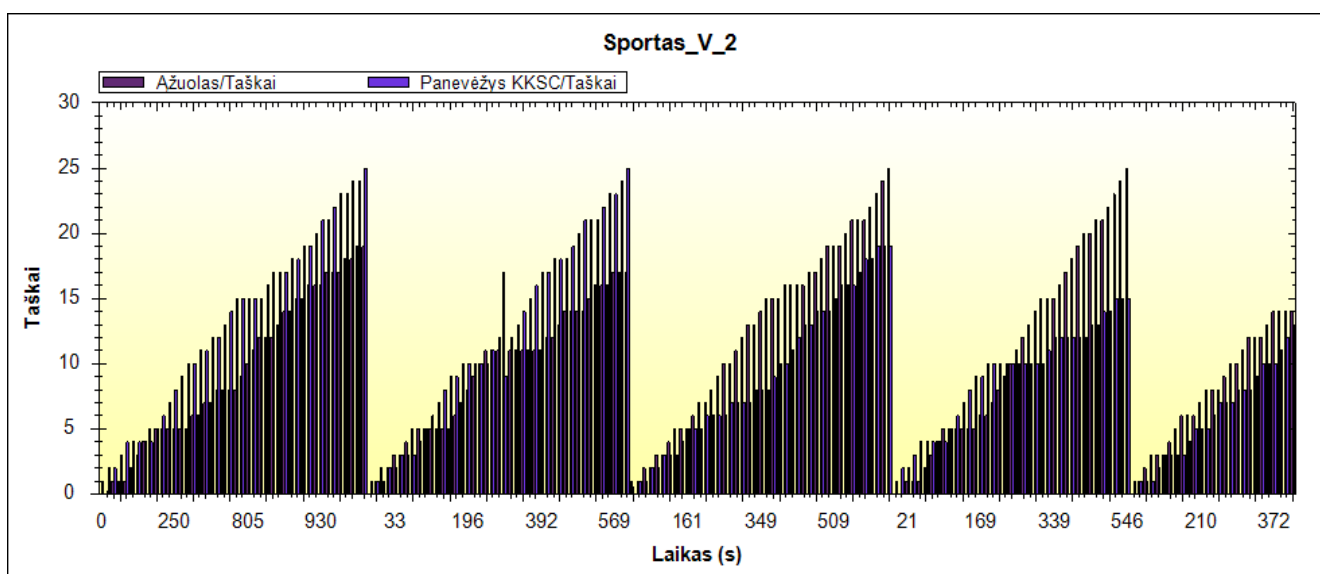
Antrame pavyzdyje pateikiama rungtynių statistika, komandų taškai. Visi duomenys paimti iš XML standartizuotų (SportML) dokumentų.

Vyrų tinklinio čempionatas 2009 - 2010

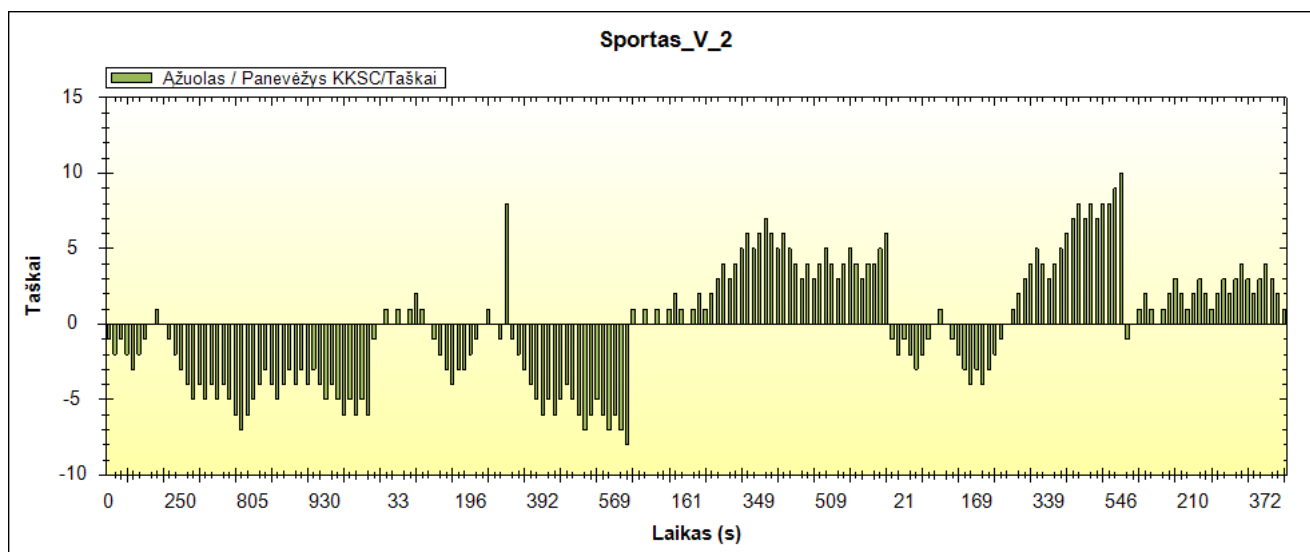
Lentelė

Data	Komanda	Taškai	Set as 1	Set as 2	Set as 3	Setas 4	Setas 5
2009.11.15	Ažuolas	101	19	17	25	25	15
2009.11.15	Panevezys KKSC	97	25	25	19	15	13

Pateikti rezultatų metu pelnyti taškai pagal setus:



Pateikti komandų setų ir taškų rezultatų skirtumai:



Lentelėje pateikiama dalis rungtynių rezultatų, kurie buvo išrinkti iš XML (SportML) dokumento:

doc-id	language	sports-title	id	key	id2	period-value	period-time-elapsed	team-idref	score-team	score-team-opposing
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1							
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1							
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	854	1	0	12	1234	1	0
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	854	1	15	20	1235	1	1
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	854	1	22	30	1234	2	1
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	854	1	35	49	1234	3	1
91201	LT	Vyrų tinklinio čempionatas	1	854	1	55	105	1235	3	2

		2009 - 2010								
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	110	121	1234	4	2
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	125	130	1234	5	2
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	135	149	1235	5	3
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	152	159	1235	5	4
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	162	170	1235	5	5
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	173	180	1235	5	6
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	185	200	1235	5	7
91201	LT	Vyrų tinklinio čempionatas 2009 - 2010	1	85 4	1	210	230	1235	5	8

