

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

TVIRTINU

Katedros vedėjas

doc. dr. D. Rubliauskas

2004 05

**MPEG-7 Standarto taikymas skaitmeniniams
vaizdams aprašyti**

Informatikos mokslo magistro baigiamasis darbas

kalbos konsultantė

lietuvių kalbos katedros lektorė

dr. J. Mikelionienė

2004 05 25

Darbo vadovas

dr. Armantas Ostreika

2004 05

Recenzentas

doc. S. Maciulevičius

2004 05

Atliko

IFM 7/3 gr. stud. Dainius Kasperiušas

2004 05

Kaunas

2004

Turinys

1. Įvadas.....	3
2. Uždavinio aplinkos analizė.....	6
2.1 Paveikslėlio formos aprašymo deskriptoriai.....	8
2.1.1 Forma srityje (<i>Contour shape</i>).....	9
2.1.2 Kontūro formos deskriptorius (<i>Region Shape</i>).....	10
2.1.3 Paveikslėlio tekstinio aprašymo principai.....	11
3. Reikalavimų specifikacija.....	12
3.1.1 Projekto užsakovas, vykdytojai ir vartotojai.....	12
3.1.2 Projekto realizavimo terminai, finansavimas, pajamos.....	13
3.1.3 Programinio produkto tikslas.....	13
3.1.4 Projekto alternatyvos.....	13
3.1.5 Programinio produkto funkcijos ir našumas.....	13
3.1.6 Projekto gyvavimo laikas.....	14
3.1.7 Vartotojų tikslai ir reikalavimai.....	14
3.1.8 Nefunkciniai reikalavimai.....	14
3.1.9 Programinės įrangos patikimumas.....	14
3.2 Projekto specifikacija.....	15
3.2.1 Projektavimo technologija.....	15
3.2.2 Naudojama programinė įranga.....	15
3.2.3 Operacinė sistema ir kita programinė įranga.....	15
3.2.4 Naudojama techninė įranga.....	15
3.3 Sistemos projektas.....	16
3.3.1 Sistemos architektūra.....	16
3.3.2 Programinio produkto galimos modifikacijos ateityje.....	16
3.4 Objektinio modelio specifikacija.....	16
Public Methods.....	19
Protected Methods.....	19
Public Methods.....	19
Protected Methods.....	20
Protected Attributes.....	20
Public Methods.....	20
Protected Methods.....	21

Protected Attributes	21
Public Methods	22
Protected Attributes	22
Public Methods	22
Protected Attributes	23
4.1 Programos langai	24
5. Testavimo specifikacija	29
5.1 Testavimo pasirinkimas	29
5.2 Automatizuota statinė analizė.....	29
5.3 Vizualinių požymių generavimo teisingumas	29
5.4 Visų galimų būsenų testavimas	29
5.5 Testavimo medžiaga.....	30
6. Rizikos įvertinimo ir mažinimo planas	31
7. Išvados	33
Literatūra.....	34
8. Summary	35
9. Terminų ir santrumpų žodynas	36

1. Įvadas

Vis daugiau ir daugiau garso ir vaizdo informacijos tampa prieinama iš daugybės šaltinių. Informacija pristatoma įvairiomis formomis; kaip paveikslų, grafikos, 3D modelių, garso, kalbos, vaizdo. Nenusakomi audiovizualinės informacijos kiekiai tampa prieinami skaitmenine forma, skaitmeniniuose archyvuose, internete, transliuojamais duomenų srautais, asmeninėse bei profesionaliose duomenų bazėse, ir šios informacijos kiekis vis didėja. Informacijos vertė dažnai priklauso nuo to, kaip lengvai ji randama, pasiekama, filtruojama ir valdoma. Per keletą ateinančių metų vartotojai susidurs su labai dideliu informacijos kiekiu, pateikiamu daugialypiais kanalais, kad efektyvus ir tikslus priėjimas prie šio beveik begalinio kiekio bus neįsivaizduojamas. Nepaisant fakto, kad vartotojai turės vis didesnę priėjimą prie šių šaltinių, atpažinti ir efektyviai juos valdyti tampa vis sudėtingiau. Ši situacija reikalauja savalaikio problemos sprendimo. MPEG-7 gali būti atsakymas į šį reikalavimą.

MPEG-7 yra ISO/IEC standartas, sukurtas MPEG (*Moving Picture Experts Group*), komiteto, kuris sukūrė sėkmingus standartus žinomus kaip MPEG-1 (1992), MPEG-2 (1995) ir MPEG-4 (1999). MPEG-7 oficialiai vadinamas "*Multimedija Content Description Interface*". MPEG-7 pateiks didelį standartizuotų įrankių rinkinį, skirtą daugialypės terpės turiniui aprašyti.

Kol garso ir vaizdo informacija vartojama tiesiogiai žmonių, vis daugėja atvejų, kur audiovizualinė informacija kuriama, pertvarkoma, atkurama ir dar kartą panaudojama skaičiavimo sistemose. Dažniausiai pasitaikantys atvejai: vaizdo suvokimas (sekimas, protingas regėjimas, gudrios kameros ir t. t.) ir turinio transformavimas (kalba transformuota į tekstą, vaizdas transformuotas į kalbą, kalba transformuota į vaizdą ir t. t.). Kiti atvejai yra informacijos išrinkimas (greita ir efektyvi informacijos atranka, susijusi su įvairiais daugialypės terpės dokumentais) ir audiovizualinio turinio srauto aprašymų, gaunamų iš daugialypės terpės atranka, atitinkanti vartotojo poreikius. Pavyzdžiui, signalas televizijos programoje įjungia tinkamai užprogramuotą vaizdomagnetofoną, kad šis įrašytą programą; vaizdo jutiklis įjungia pavojaus signalą, kai įvyksta tam tikras įvykis. Automatinis iškodavimas gali būti atliekamas iš ženklų eilutės į garsinę informaciją, arba paieška gali būti atliekama garso ar vaizdo duomenų sraute. Visuose šiuose pavyzdžiuose audiovizualinė informacija turi būti tinkamai užkoduota, kad priverstų įrenginį ar kompiuterinį kodą įvykdyti užprogramuotą veiksmą.

MPEG-7 standartas kuriamas, atsižvelgiant ir į kitus standartus, kaip PCM, MPEG-1, -2 ir -4. Viena MPEG-7 standarto funkcijų yra pateikti ryšius į atitinkamas jų dalis. Pavyzdžiui, kontūro deskriptorius naudojamas MPEG-4 yra naudingas MPEG-7 kontekste, taip pat kaip ir judesio vektorių laukai naudojami MPEG-1 ir MPEG-2.

MPEG-7 leidžia turėti skirtingus aprašymų detalizavimo lygius ir siūlo galimybę turėti skirtingus traktavimo lygius. Nors MPEG-7 aprašymai nepriklauso nuo turinio pateikimo (kodavimo), MPEG-7 gali išnaudoti MPEG-4 užkoduoto turinio privalumus. Jei medžiaga užkoduota naudojant MPEG-4, šiuo būdu galima užkoduoti audiovizualinę informaciją kaip objektus, turinčius tam tikrus ryšius laike (sinchronizacija) ir erdvėje (vaizdui ekrane ir garsui patalpoje). Yra galimybė susieti aprašymus su elementais (objektais) scenoje, pvz. garso ar vaizdo objektais.

MPEG-7 aprašymo įrankiai leis sukurti turinio aprašymus, kurie apims [1]:

- Informaciją, apibūdinančią turinio kūrimo ir gamybos procesą (režisierius, pavadinimas, trumpas aprašymas).
- Informaciją, susijusią su turinio naudojimu (autorinių teisių rodyklės, naudojimo istorija, transliavimo laikas).
- Informaciją apie turinio saugojimo ypatybes (saugojimo formatas, kodavimas).
- Struktūrinę informaciją apie turinio erdvinius, laiko ir erdvės-laiko komponentus (scenų pasikeitimai, regionų segmentacija, regionų judėjimo sekimas).
- Informaciją apie turinio žemo lygio savybes (spalvos, tekstūros, garso tundrai, melodijos aprašymas).
- Abstrakčią informaciją - realybės užfiksuotos turinyje (objektai ir įvykiai, sąveika tarp objektų).
- Informaciją, kaip efektyviai žiūrėti turinį (santraukos, pokyčiai).
- Informaciją apie objektų rinkinius.
- Informaciją apie vartotojo sąveiką su turiniu (vartotojo pageidavimai, naudojimo statistika).

Visi šie aprašymai užkoduoti efektyviu būdu filtravimui, paieškai ir kt.

MPEG-7 duomenys gali būti kartu su garso ir vaizdo informacija – tai yra tame pačiame duomenų sraute, bet gali būti laikomi ir visai atskirai. Kai *meta* duomenys ir duomenys laikomi atskirai, būtinas mechanizmas, kuris sujungtų garso ir vaizdo informaciją su aprašu.

MPEG-7 standartas naudoja XML kalbą arba dvejetainį aprašymo būdą. XML kalba naudojama dėl jos vartojimo populiarumo bei siekiant palengvinti uždavinio sprendimą ateityje.

Pagrindiniai MPEG-7 standarto elementai yra:

Deskriptorius ir deskriptorių schema, kuri apibūdina ryšių tarp komponentų semantiką ir struktūrą.

MPEG-7 yra pritaikytas tiek realaus laiko aplinkai tai yra – garsas ir vaizdas gali būti tuo metu transliuojamas arba archyvuose saugomai informacijai aprašyti.

MPEG-7 Visual aprašymų įrankiai susideda iš pagrindinių struktūrų ir deskriptorių, kurie aprėpia tokias elementarias vaizdo sekos savybes[2]:

- Spalvą
- Tekstūrą
- Formą
- Judėjimą
- Lokalizavimą
- Kitas

2. Uždavinio aplinkos analizė

Pagrindinis darbo uždavinys: Išsiaiškinti MPEG-7 standartą ir (pritaikyti konkreitiems uždaviniams spręsti).

Tam, kad galėtume kurti programinės įrangos projektą, pirmiausiai turime išsianalizuoti jau esamą PĮ, ištirti jos privalumus, trūkumus bei galimybes.

Žemiau pateikiami kelių profesionalių ir eksperimentinių panašaus tipo bei daugiau mažiau analoginių programinių paketų trumpi aprašymai, parodomi pagrindiniai jų privalumai ir trūkumai;

Annotation Tool

Gamintojas: IBM

Operacinė sistema: Win9x, WinNT, Win2000, WinME, WinXP.

Palaikomi bylų formatai: MPEG (*.mpg).

Aprašymas:

Eksperimentinė programa, padedanti aprašyti vaizdo seką MPEG-7 standartu. Aprašant vaizdo seką MPEG-7 standartu, vaizdo failas sukarpomamas mažomis atkarpėlėmis – paveikslėliais- ir tose atkarpėlėse matomi įvykiai, objektai. Paveikslėliai aprašomi atskirai ir išsaugomi kaip MPEG-7 aprašai XML faile. Programa taip pat gali atidaryti MPEG-7 failą ir parodyti vaizdo segmentų aprašus. Svarbu pabrėžti, kad programa leidžia aprašyti bet koki įvykį, susikurti specifinius įvykius, t. y. kurti naujus leksikonus arba užkrauti vartotojo išsaugotus [3].

Sistemos privalumai ir galimybės:

- Draugiška aplinka, aiškus programos valdymas.
- Galima kurti naujus įvykius arba pasirinkti iš programoje numatytos įvykių aibės.
- Sistema automatiškai sukarpo vaizdo failą atkarpomis.
- Objektų padengimas tekstūromis.

Sistemos trūkumai:

- Pritaikyta tik vienam formatui *.mpg.
- Sistema eksperimentinė, produktas neišbaigtas.
- MPEG-7 aprašas formuojamas žmogaus pagalba, o ne automatiškai.

- Nėra sukurta paieškos sistema.

Ricoh MovieTool

Gamintojas: Ricoh

Operacinė sistema: Win2000.

Palaikomi bylų formatai: MPEG-1.

Aprašymas:

Ši sistema skirta vaizdo failams aprašyti MPEG-7 formatu. Naudojant sistemą, galima ne tik kurti MPEG-7 failus, bet ir juos redaguoti. Tam skirta didelė modifikavimo priemonių gama. Patogi vartotojo sąsaja su programa leidžia greitai ir patogiai redaguoti failus ir juos peržiūrėti[4].

Sistemos privalumai ir galimybės:

- Sukuria MPEG-7 aprašą, įkėlus vaizdo failą.
- Patogus ir interaktyvus failų kūrimas bei modifikavimas.
- Tikrina, ar sukurtas failas atitinka MPEG-7 standarto reikalavimus.

Sistemos trūkumai:

- Kitų sistemų sukurti MPEG-7 failai gali būti atpažinti neteisingai.
- Sistema apdoroja tik MPEG-1 kodo failus.
- Sistema nepalaiko paieškos funkcijos MPEG-7 standarte.
- Sistema veikia tik WIN 2000 aplinkoje.
- Nepalaiko dvejetainio MPEG-7 formato.
- Sistema yra komercinė.

Canon MPEG-7 Spoken Content Transcription Service

Gamintojas: Canon

Palaikomi bylų formatai: .wav.

Aprašymas:

Canon firmos tyrimų laboratorija sukūrė šnekamosios kalbos aprašymo programą. Programa gali pasinaudoti visi interneto vartotojai; jie turi nusiųsti garsinį failą, programa sukurs ir gražins XML failą, atitinkantį MPEG-7 *SpokenContent* deskriptoriaus schemą. Šiuo metu sistema atpažįsta tik fonemas (neatpažįsta žodžių) ir gali dirbti tik su riboto dydžio .wav failais [5].

Singingfish Multimedija Search

Gamintojas: Singingfish

Operacinė sistema: Win9x, WinNT, Win2000, WinME, WinXP, Linux.

Palaikomi bylų formatai: Microsoft Windaus Media, Apple QuickTime, MP3, RealNetwork, RealVidio, RealAudio bei Real Flash.

Aprašymas:

Singingfish savo paieškos sistemoje naudoja paieškos variklį, skirtą srautinės multimedijos (*streaming media*) paieškai. Paieškos variklis naudoja MPEG-7 Multimedija Description Schemes ir atlieka internete esančios *streaming media* informacijos paiešką. Terminu „streaming media“ čia vadiname visus internete esančius multimedia duomenis, kuriuos galima peržiūrėti arba klausytis dar nebaigus parsisiųsti failo. Singingfish savo duomenų bazėje turi nuorodas į Microsoft Windaus Media, Apple QuickTime, MP3 ir RealNetwork, RealVidio, RealAudio bei Real Flash formatu [6].

Sistemos privalumai ir galimybės:

- Paieška dirba su daugeliu populiariausių multimedija formatų.

Sistemos trūkumai:

- Paieškos sistema nedirba su paveikslėlių formatais, tokiais kaip .jpg .bmp ir kt.
- Singingfish naudojamu paieškos varikliu galima naudotis tik Internete.

2.1 Paveikslėlio formos aprašymo deskriptoriai

Formos deskriptorius galima būtų suskirstyti i tris grupes [7]: 2-dimensijų(2D), 3-dimensijų (3D) ir tekstinį laisvą formos aprašymą.

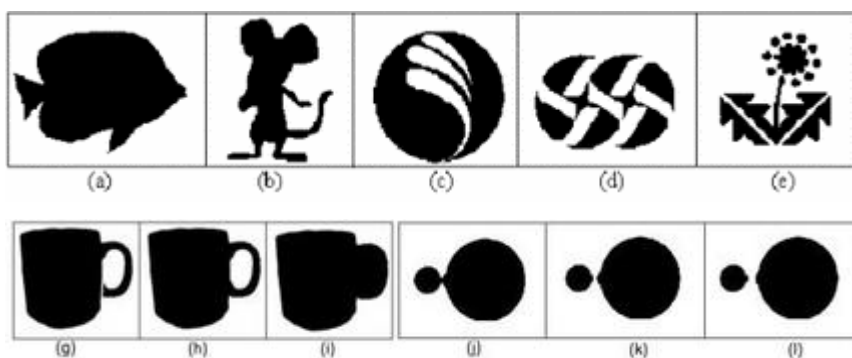
2D formai aprašyti naudojami du deskriptoriai, kurie charakterizuoja skirtingus 2D objektų ar kontūrų požymius. **Region Shape** deskriptorius nuskaito visus regiono pikselius ir tada aprašo formą, o **Contour Shape** deskriptorius charakterizuoja formos savybes.

3D formos deskriptoriai gali būti aprašomi dviem būdais. Naudojant (*Shaape3D*) deskriptoriaų objekto tikroji forma nustatoma apgaubiant 3D modelį tinkleliu. Alternatyvus metodas taikomas aprašyti 3D objektams tai- (*MultipleView*) deskriptoriaus panaudojimas kartu su 2D formos deskriptoriumi. Metodas patogus, kai objekto 3D tinklelis nežinomas arba kai reikalingas objekto 2D vaizdas.

Kuriant paveikslėlio tekstinį aprašą, nenaudojamos jokios automatizavimo sistemos, išskyrus iš anksto sukurtus šablonus darbo palengvinimui.

2.1.1 Forma srityje (*Contour shape*)

Šiuo atveju aprašomą formą gali sudaryti vienas ar daugiau kontūrų. Formoje gali būti išpjovos (tokio tipo formas mes matome paveikslėlyje 1). Kadangi deskriptorius nagrinėja visą sritį ir naudoja visus pikselius, jis gali aprašyti norimą formą t. y. ne tik paprastas formas, sudarytas iš vieno kontūro, kaip matome paveikslėlyje 1 (a) ir (b), bet ir sudėtingas formas su išpjovomis arba tokias, kurias sudaro keli kontūrai 1 pav. (c), (d) ir (e). Deskriptorius patogus, nes gali aprašyti įvairiausias formas minimaliai iškraipydamas jos pakraščius.

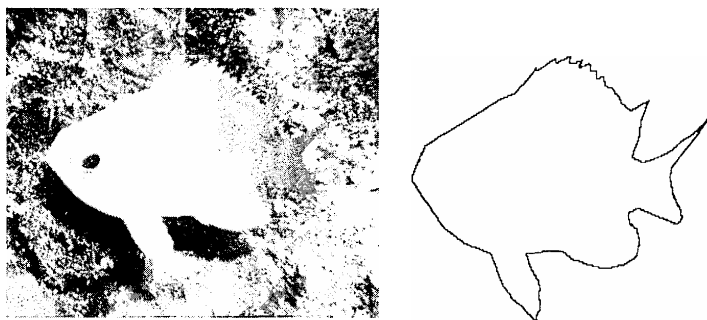


1 pav. Skirtingų formų pavyzdžiai

Pav.1 (g), (h) ir (i) vaizduojami labai panašūs objektai. Puodelis pav. 1 (g) turi įtrūkimą rankenoje, o puodelio (i) rankena iš viso be išpjovos. Formos deskriptorius, naudojantis visą aprašomą sritį (g) ir (h) palaikys panašiais, o (i) skirtingu nuo pirmų dviejų. Panašiais deskriptorius pripažins ir vaizdus, matomus (j-l). Čia vaizduojama vaizdo kadru seką, kurioje du diskai po truputį tolsta vienas nuo kito. Formos sritį aprašančio deskriptoriaus savybėgreita paieška, o jo sudarytas aprašas yra nedidelis ir užima fiksuota 17,5 baito dydį.

2.1.2 Kontūro formos deskriptorius (*Region Shape*)

Deskriptorius aprašo uždara 2D objektą ar regioną paveikslėlyje arba vaizdo sekoje 2 pav.



2 pav. Uždaras regionas paveikslėlyje ir jį atitinkantis kontūras.

Formai aprašyti deskriptorius naudoja *Curvature Scale Space* (CSS) principą. Šiuo būdu gaunamas aprašymas yra labai kompaktiškas ir užima vidutiniškai mažiau nei 14 baitų. Contour shape – XML aprašo sintaksė turėtų atrodyti maždaug taip[]:

DDL representation syntax

```
<complexType name="RegionShapeType" final="#all">
  <complexContent>
    <extension based="mpeg7:VisualDType">
      <sequence>
        <element name="MagnitudeOfART">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="mpeg7:unsigned4"/>
              </simpleType>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

(Region Shape) deskriptoriaus privalumai ir ypatybės:

- Gaunamos puikios formos charakteristikos leidžiančios vėliau atlikti panašių formų paiešką.

- Gaunama forma atspindi žmogaus akies sukuriama formos vaizdą.
- Labai kompaktiška.

Region Shape trūkumai:

- Gan sudėtinga realizuoti.

Paveikslėlio formos aprašymo deskriptorių galimas taikymas:

- virtualių erdvinių pasaulių aprašymui;
- duomenų kaupimo sistemose;
- internete ir panašiai.

2.1.3 Paveikslėlio tekstinio aprašymo principai

Kaip jau buvo minėta anksčiau, MPEG-7 leidžia du aprašymo lygius: automatizuotą ir rankinį. Aukščiau nagrinėti „Konturo formos“ ir „Forma srityje“ aprašai buvo gaunami automatizuotu būdu. Dabar panagrinėsime, kaip turi atrodyti rankiniu būdu aprašytas mpeg-7 failo formatas, dar kitaip šis aprašas vadinamas semantiniu.

Žemiau pateiktas pavyzdys aprašo vaizdo sceną, kurioje matome kaip Jonas meta kamuolį Petriui, o Petras pagauna kamuolį kaire ranka.

```
<MPEG7>
  <Description xsi : type="ContentEntityType">
    <MultimediaContent xsi : type="VideoType">
      <Video>
        <MediaLocator>
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation xml : lang = "en">
            Jonas throws a small ball to Petras.
            Petras catches the ball with his left hand.
          </FreeTextAnnotation>
          <StructuredAnnotation>
            <Who href="urn : people : Jonas100">
              <Name xml : lang = "en">Jonas
              </Name>
            </Who>
            <Who>
              <Name xml : lang = "en">Petras
              </Name>
            </Who>
            <WhatObject>
              <Name xml : lang = "en"> A small ball
              </Name>
            </WhatObject>
          </StructuredAnnotation>
        </TextAnnotation>
      </Video>
    </MultimediaContent>
  </Description>
</MPEG7>
```

```

</WhatObject>
<WhatAction>
  < Name xml : lang = " en " > Jonas
throws a small ball to Petras.
  </Name>
</WhatAction>
<WhatAction>
  < Name xml : lang = " en " > Petras
catches the ball with his left hand.
  </Name>
</WhatAction>
</StructuredAnnotation>
</TextAnnotation>
</Video>
</MultimediaContent>
</Description>
</MPEG7>

```

Pavyzdyje naudojami du tekstinio aprašo būdai:

- a. laisvas tekstinis aprašas (*Free Text Annotation*);
- b. Struktūrizuotas tekstinis aprašas (*Structured Annotation*).

Struktūrizuoto tekstinio aprašymo atveju *Jonas* identifikuojamas kaip *Who*, o metamas kamuolys, kaip *What*. *Jonas* yra identifikuojamas, naudojant terminologiją "urn : people : Jonas100", kuri yra naudojama *people* klasifikavimo schemeje.

Apibendrinami išnagrinėtos programinės įrangos ir paveikslėlių aprašymo įrankių privalumus ir trūkumus, iškeliamė sau uždavinį sudaryti programinės įrangos projektą, kuriame įvertinsime jau esamų sistemų privalumus ir papildomas galimybes, stengsimės išvengti jų trūkumų, arba pasirinkti optimaliausią variantą, įvertinant būsimų klientų poreikius ir savo galimybes.

3. Reikalavimų specifikacija

3.1.1 Projekto užsakovas, vykdytojai ir vartotojai

Projekto užsakovas ir vadovas yra Kauno technologijos universiteto Informatikos fakulteto Praktinės informatikos katedros dėstytoja Erinija Pranckevičienė.

Projektą vykdo Kauno Technologijos Universiteto Informatikos fakulteto Informacinių Technologijų magistrantas Dainius Kasperiušas.

Programinio produkto vartotojai:

- Gydytojų įstaigos, turinčios kompiuterinės tomografijos laboratorijas ir atliekančios žmogaus smegenų kompiuterinės tomografijos tyrimus;
- Profesionalai ir mėgėjai, dirbantys su nedideliais vaizdines informacijos kiekiais;
- Asmeninių kompiuterių vartotojai, siekiantys susisteminti savo skaitmeninių vaizdų archyvus.

3.1.2 Projekto realizavimo terminai, finansavimas, pajamos

Programinis produktas, atitinkantis visus vartotojų reikalavimus, numatomas realizuoti iki 2004 metų antro ketvirčio pabaigos (gegužės – birželio mėn.).

Programinio produkto kūrimui finansavimas nenumatytas. Tačiau iš dalies projekto kūrimą remia Kauno technologijos universitetas, suteikdamas teisę nemokamai naudotis ISO/IEC 15938 standarto dokumentacija.

Kadangi projektas kuriamas akademiniais tikslais, pajamos iš programinio produkto vartotojų nenumatytos.

3.1.3 Programinio produkto paskirtis

Programinis produktas – informacinė sistema, kurios pagrindinė funkcija vaizdinės informacijos kodavimas/dekodavimas į XML standarto pagrindu sukurtą vizualinės informacijos formatą.

3.1.4 Projekto alternatyvos

Projektui įgyvendinti galimi ir alternatyvūs būdai, pavyzdžiui: informacijos, išgautos iš kompiuterinių tomografų, saugojimui galima naudoti savo susikurtą duomenų formatą. Tačiau nuspręsta vaizdinės informacijos saugojimui naudoti MPEG-7 standartą, atitinkantį informacijos saugojimo formatą. Šis standartas yra sukurtas audiovizualinės informacijos aprašymui bus labai plačiai ir perspektyviai naudojamas netolimoje ateityje.

Plačiau su MPEG-7 standartu galima susipažinti [2].

3.1.5 Programinio produkto funkcijos ir našumas

Programinio produkto funkcijos:

- Duomenų nuskaitymas iš šių formatų grafinių failų (BMP, JPG)
- Pažymėtų grafinių failų aprašymas MPEG-7 formatu
- MPEG-7 formato failų dekodavimas ir paieška pagal pasirinktus požymius (tekstinį aprašą)

3.1.6 Projekto gyvavimo laikas

Numatomas programos gyvavimo laikotarpis 2 metai, kadangi manoma, jog per šį laikotarpį atsiras komercinių produktų, atliekančių šias funkcijas patikimiau ir turinčių daugiau galimybių.

3.1.7 Vartotojų tikslai ir reikalavimai

Vartotojų pageidavimai – paprastas ir patogus programos naudojimas. Lengva ir greita išsaugotos informacijos paieška duomenų archyve. Visos programos funkcijos turi būti aiškios ir lengvai įvaldomos.

Vartotojo sąsaja su programa turi būti suprantama, neperkrauta, lengvai valdoma, patogi, su lengvai pasiekiamą pagalba, maksimaliai orientuota profesinei vartotojo aplinkai.

3.1.8 Nefunkciniai reikalavimai

Išnagrinėjus panašaus tipo programinę įrangą bei jos reikalavimus sistemoms, planuojami programinės bei aparatūrinės įrangos reikalavimai būtų tokie:

Aparatūrinė įranga:

IBM PC.

Programinė įranga:

- Programos greitis priklausys nuo turimos techninės įrangos.
- Planuojamas programos dydis: iki 10Mb.
- OS: Windows 9x, ME, NT, 2000, XP.

Viena iš labiausiai paplitusių operacinių sistemų.

Programavimo kalba: C++ (programinis produktas MS Visual .NET sukompiliuota su MFC bibliotekų 7.10.3077.0 versija), tai - daugiausiai bibliotekų turinti bei viena populiariausių programavimo kalbų. Lanksti, palaiko objektinį programavimą.

Planuojama reikalinga techninė įranga: Min: P1, 800Mb HDD, 64 RAM,

3.1.9 Programinės įrangos patikimumas

Programinis paketas turi veikti stabiliai, neturi vyksti sistemos trikdžių, atliekant uždavinius, nuskaitant informaciją ar norint ją išsaugoti. Sistema turi veikti visiškai patikimai nuo pat darbo su ja pradžios iki galo.

3.2 Projekto specifikacija

3.2.1 Projektavimo technologija

Projektuojant mūsų sistemą, buvo pasirinktas *Krioklio* modelis, kadangi yra aiškiai aprašyti ir gerai suprantami kuriamos sistemos reikalavimai. Yra išsiaiškinta su vartotoju, kokia turi būti sistema. Sistemos projektavimas bus vykdomas remiantis *Objektinio projektavimo* principu.

3.2.2 Naudojama programinė įranga

Programavimo kalba

Pasirinkta objektiškai orientuota programavimo kalba c++, kadangi numatomas sistemos objektinis projektavimas. C++ pasirinkta ir todėl, kad parašytas programos galima nesunkiai pritaikyti įvairioms operacinėms sistemoms.

Automatizavimo priemonės

Programavimo automatizavimui bus panaudotas programinis produktas MS Visual .NET.

Tai programų kūrimo paketas, skirtas palengvinti rašyti programas c++ programavimo kalboje.

Programos derinimo priemonės

Derinimas vykdomas atsižvelgiant į kompiliavimo metu išvedamus klaidų pranešimus. Išvedami klaidų pranešimai pakankamai informatyvūs – nurodomas eilutės numeris, kurioje aptikta sintaksės ar kitokio pobūdžio klaida. Tiksliai nurodomas kintamojo ar klasės pavadinimas, kurių gyvavimo lauke aptikta klaida.

3.2.3 Operacinė sistema ir kita programinė įranga

Visi projektavimo ir programavimo darbai atliekami Microsoft Windows operacinių sistemų šeimoje.

Dokumentavimui naudojamas Microsoft Word 2000 tekstinis redaktorius.

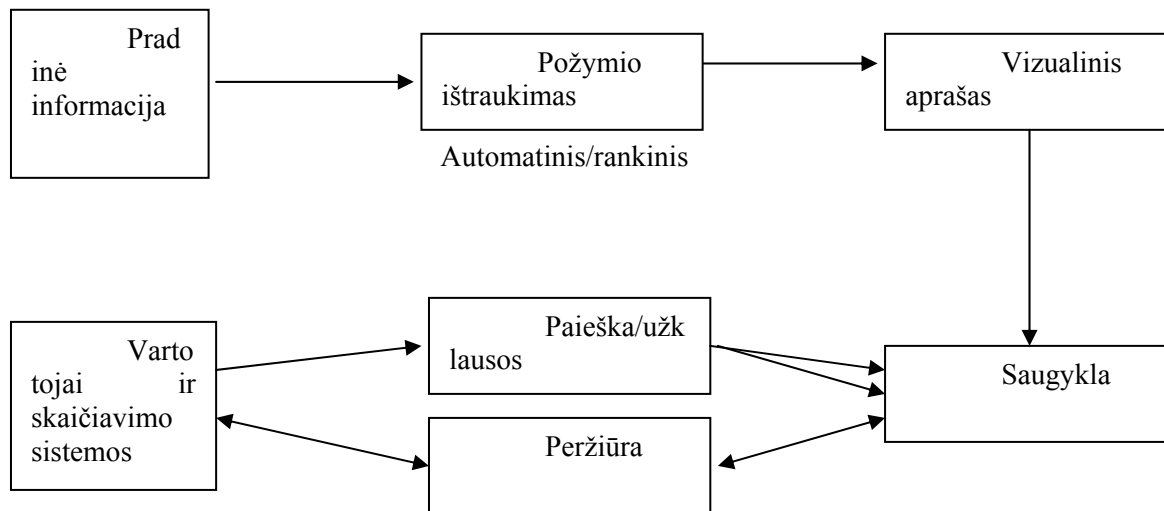
3.2.4 Naudojama techninė įranga

Asmeninis kompiuteris: Procesorius AMD Athlon XP 1.7 GHz

Operatyvinė atmintis: 256 MB

3.3 Sistemos projektas

3.3.1 Sistemos architektūra



3 pav. Sistemos architektūros schema.

Pradinė informacija – BMP arba JPG formato grafinis failas.

Požymio ištraukimas – tekstinis duomenų ištraukimas.

Saugykla – vizualinių aprašų duomenų bazė saugoma pagal nuožiūrą pasirinktame kataloge.

Paieška/užklauskos – vartotojo įvedamos užklauskos.

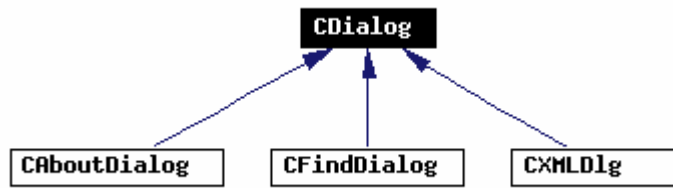
Peržiūra – programinis įrankis archyvo peržiūrai.

3.3.2 Programinio produkto galimos modifikacijos ateityje

Ateityje užsakovo prašymu bus galima išplėsti programinio produkto galimybes naujais įvedamų duomenų formatais (pvz. GIF, PNG ir .t.t.) ir papildomais požymiais.

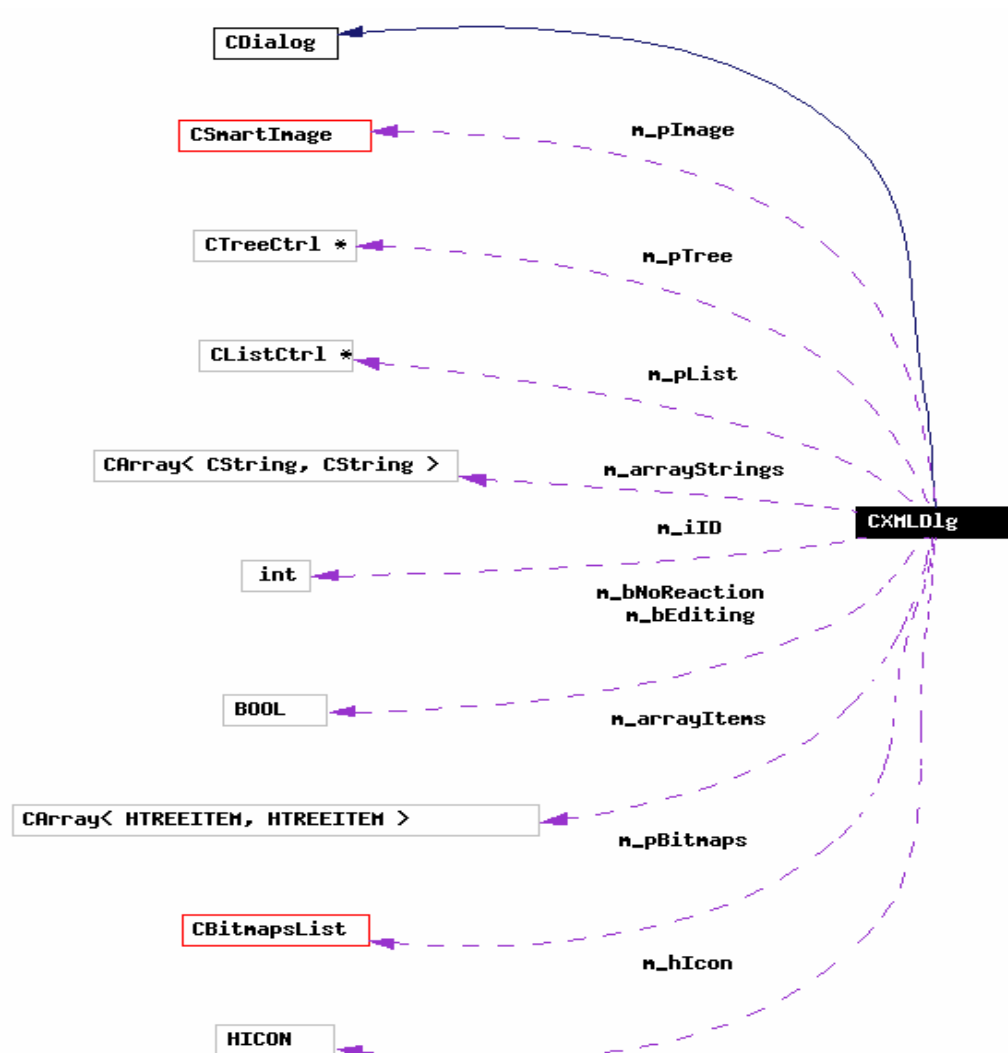
3.4 Objektinio modelio specifikacija

Kadangi pagrindinis šios programos darbas – bendrauti su vartotoju (programos tikslas – palengvinti ir automatizuoti XML failo generavimą), programos objektinė struktūra koncentruojasi aplink pagrindinius vartotojo sąsajos objektus – dialogo langus. Jų šioje programoje yra 3 4. pav. Vartotojo sąsajos langai:



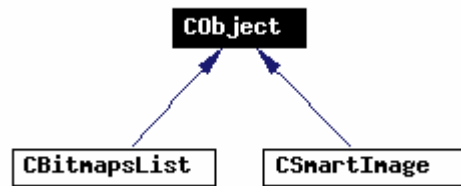
4. pav. Vartotojo sąsajos langai

Programos pagrindas – CXMLDlg klasė 5. pav., joje apdorojami dialogo, atstovaujamo pagrindinį programos langą, įvykiai, bei vyksta konvertavimas iš XML į vidinius programos duomenis ir atvirkščiai.



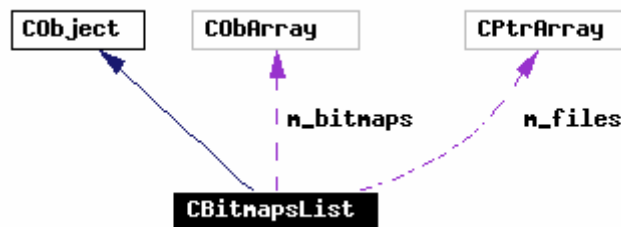
5. pav. CXMLDlg klasė

Kaip matome, ši klasė inkapsuliuoja visą paletę objektų, duomenų struktūrų ir tipų: MFC standartinius CArray, CTreeCtrl, CViewCtrl bei naujus CSmartImage bei CbitmapsList 6.pav.



6.pav. Klasių inkapsuliacija.

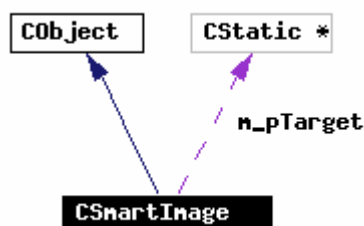
CBitmapsList paskitis - saugoti piešinukų sąrašą bei failų pavadinimus. Viena iš papildomų klasės CBitmapsList paskirčių yra CImageList objekto generavimas iš saugomų



7.pav. Klasė CbitmapsList.

piešinukų sumažintų kopijų 7.pav.

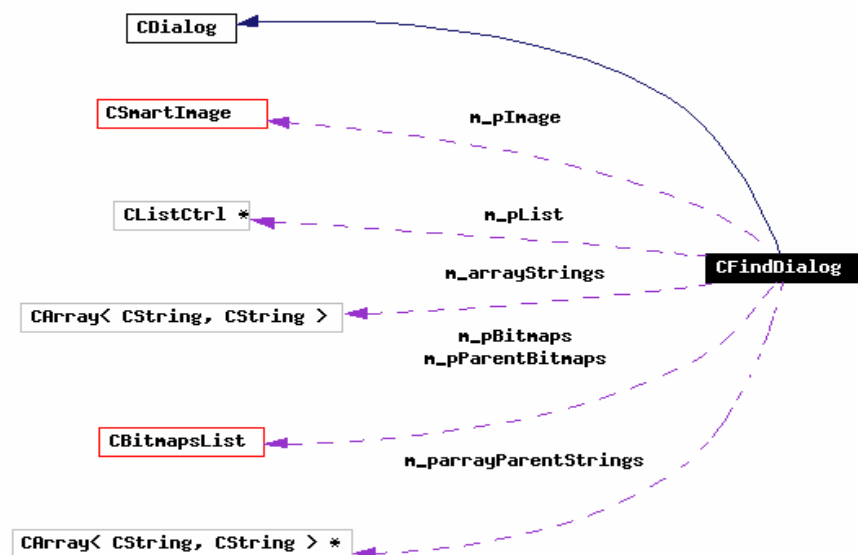
CSmartImage - nedidelė, pagalbinė klasė, skirta piešinuko projektavimui į nurodytą kliento lango srit 8.pav. Ši klasė sugeba nusistatyti optimalų didinimo/mažinimo algoritmą ir



8.pav. Klasė CSmartImage

suprojektuoti vizualų piešinuko vaizdą į nurodytą lango kliento sritį.

CBitmapsList ir CImage klasės naudojamos ir paieškos dialoge.



9. pav. CFindDialog klasė.

CFindDialog klasė perima dalį pagrindinio lango funkcionalumo ir leidžia vartotojui efektyviai ieškoti piešinių, bei peržiūrėti surastus 9. pav.

Išplėstinė informacija apie klases:

CAboutDialog

Public Methods

[CAboutDialog](#) (CWnd *pParent=NULL)
 virtual [~CAboutDialog](#) ()

Protected Methods

virtual void [DoDataExchange](#) (CDataExchange *pDX)

10. pav. CAboutDialog

CFindDialog

Public Methods

[CFindDialog](#) (CWnd *pParent, [CBitmapsList](#) *pParentBitmaps, CArray< CString, CString > *parrayParentStrings)
 virtual [~CFindDialog](#) ()

afx_msg void [OnPaint](#) ()
afx_msg void [OnBnClickedFindfind](#) ()
afx_msg void [OnLvnItemActivateFindcatalog](#) (NMHDR *pNMHDR, LRESULT *pResult)
virtual BOOL [DestroyWindow](#) ()
afx_msg void [OnBnClickedFindview](#) ()

Protected Methods

void [Clear](#) ()
void [MoveItemToChild](#) (int iItemNr)
virtual void [DoDataExchange](#) (CDataExchange *pDX)
virtual BOOL [OnInitDialog](#) ()

Protected Attributes

[CSmartImage](#) * [m_pImage](#)
[CBitmapsList](#) * [m_pBitmaps](#)
CListCtrl * [m_pList](#)
[CBitmapsList](#) * [m_pParentBitmaps](#)
CArray< CString, CString > * [m_parrayParentStrings](#)
CArray< CString, CString > [m_arrayStrings](#)

11. pav. CFindDialog

CXmldlg

Public Methods

[CXMLDlg](#) (CWnd *pParent=NULL)
afx_msg void [OnBylaBaigtiDarba](#) ()
afx_msg void [OnBylaPridetiPiesinuka](#) ()
afx_msg void [OnBnClickedAdd](#) ()
afx_msg void [OnBnClickedAddmain](#) ()
afx_msg void [OnBnClickedDelete](#) ()
afx_msg void [OnTvnEndlabeleditLexicon](#) (NMHDR *pNMHDR, LRESULT *pResult)
afx_msg void [OnTvnBeginlabeleditLexicon](#) (NMHDR *pNMHDR, LRESULT *pResult)
afx_msg void [OnBnClickedEdit](#) ()

afx_msg void [OnLvnItemActivateCatalog](#) (NMHDR *pNMHDR, LRESULT *pResult)

afx_msg void [OnBnClickedAddpicture](#) ()

afx_msg void [OnBnClickedRemovepicture](#) ()

afx_msg void [OnLvnInsertitemCatalog](#) (NMHDR *pNMHDR, LRESULT *pResult)

virtual BOOL [DestroyWindow](#) ()

afx_msg void [OnClose](#) ()

afx_msg void [OnNMClickLexicon](#) (NMHDR *pNMHDR, LRESULT *pResult)

afx_msg void [OnTimer](#) (UINT nIDEvent)

afx_msg void [OnEnChangeFreeanontation](#) ()

afx_msg void [OnPaieska](#) ()

afx_msg void [OnApie](#) ()

afx_msg void [OnIrasytiZodyna](#) ()

afx_msg void [OnBylaSkaitytiZodyna](#) ()

afx_msg void [OnBylaIrasytiKataloga](#) ()

afx_msg void [OnBnClickedView](#) ()

afx_msg void [OnBylaSkaitytiKataloga](#) ()

Protected Methods

virtual void [DoDataExchange](#) (CDataExchange *pDX)

void [CheckIfText](#) (CString &item)

void [ResetCheckBoxes](#) ()

void [GetItemsArray](#) (HTREEITEM hItem)

void [FormTerms](#) (MSXML::IXMLDOMDocumentPtr &xmlDoc, MSXML::IXMLDOMElementPtr &xmlItem, HTREEITEM hItem, CString &indent)

void [ReadTerms](#) (MSXML::IXMLDOMElementPtr &xmlItem, HTREEITEM hItem)

void [FormMPEG7](#) (MSXML::IXMLDOMDocumentPtr &xmlDoc, MSXML::IXMLDOMElementPtr &xmlItem, CString &indent)

virtual BOOL [OnInitDialog](#) ()

afx_msg void [OnPaint](#) ()

virtual void [OnOK](#) ()

virtual void [OnCancel](#) ()

afx_msg HCURSOR [OnQueryDragIcon](#) ()

Protected Attributes

HICON [m_hIcon](#)

[CSmartImage](#) * [m_pImage](#)

[CBitmapsList](#) * [m_pBitmaps](#)

CListCtrl * [m_pList](#)
 CTreeCtrl * [m_pTree](#)
 BOOL [m_bEditing](#)
CArray< HTREEITEM, HTREEITEM > [m_arrayItems](#)
 CArray< CString, CString > [m_arrayStrings](#)

12. pav. CXmlDlg

CBitmapsList

Public Methods

[CBitmapsList](#) ()
virtual [~CBitmapsList](#) ()
 void [AddBitmap](#) (CBitmap *pBmp, TCHAR *fileName)
CBitmap * [GetBitmap](#) (INT_PTR iIndex)
CBitmap * [GetSmallBitmap](#) (INT_PTR iIndex)
CImageList * [GetImagesList](#) ()
 INT_PTR [GetCount](#) ()
 void [RemoveBitmap](#) (INT_PTR iIndex)
TCHAR * [GetFile](#) (INT_PTR iIndex)

Protected Attributes

CObArray [m_bitmaps](#)
CPtrArray [m_files](#)

13. pav. CBitmapsList

CSmartImage

Public Methods

[CSmartImage](#) ()
 [CSmartImage](#) (CStatic *pStatic)
virtual [~CSmartImage](#) ()

```
void SetTarget (CStatic *pStatic)  
CStatic * GetTarget ()  
void Paint (CBitmap *pBmp)
```

Protected Attributes

```
CStatic * m\_pTarget
```

14. pav. CSmartImage

Failų sistemos ir programos sąsaja:

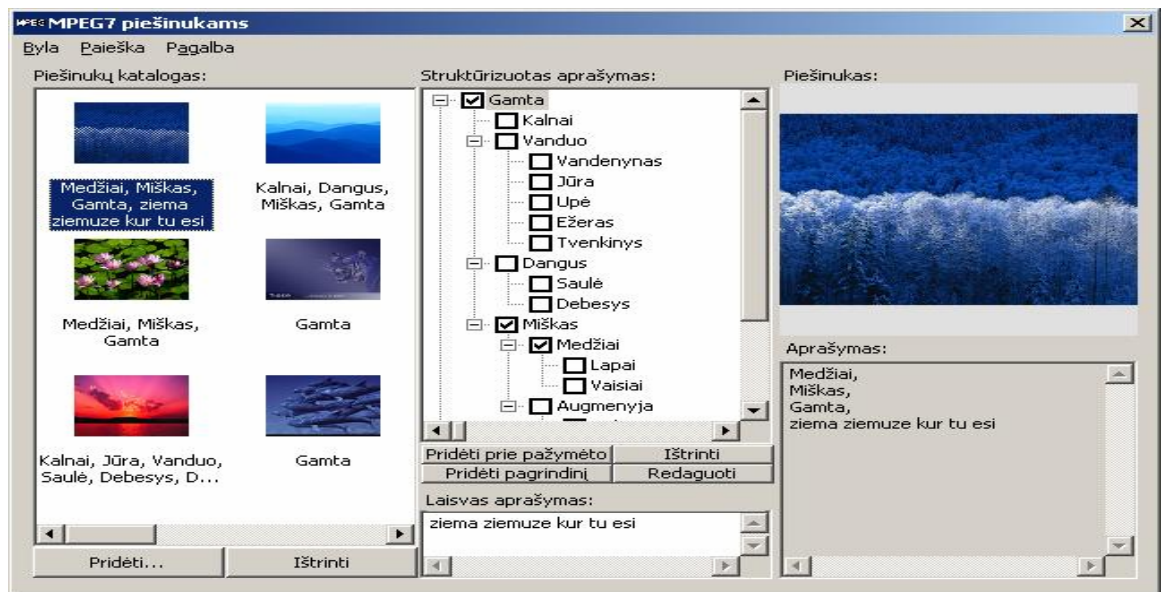
Programa bendrauja su failų sistema netiesiogiai. Piešinukais pasirūpina CImage objektas, o XML failai generuojami MSXML (sisteminė biblioteka, atsakanti už operacijas su XML dokumentais, objektais, elementais atributais ir t.t.) pagalba. Kodas, generuojantis XML iš programos duomenų ir iš XML kodo generuojantis programos duomenis inkapsuluotas CXmlDlg klasėje, nes yra stipriai priklausomas nuo CXmlDlg klasės duomenų, bei šiame dialoge esančių langų.

4 Programos P.A.S.-mpeg7 vadovas

4.1 Programos langai

Pagrindinis programos langas

Paleidus programą P.A.S.-mpeg7 atsidaro pagrindinis programos langas, parodytas 15.pav.



15. pav. Pagrindinis programos langas

Meniu.

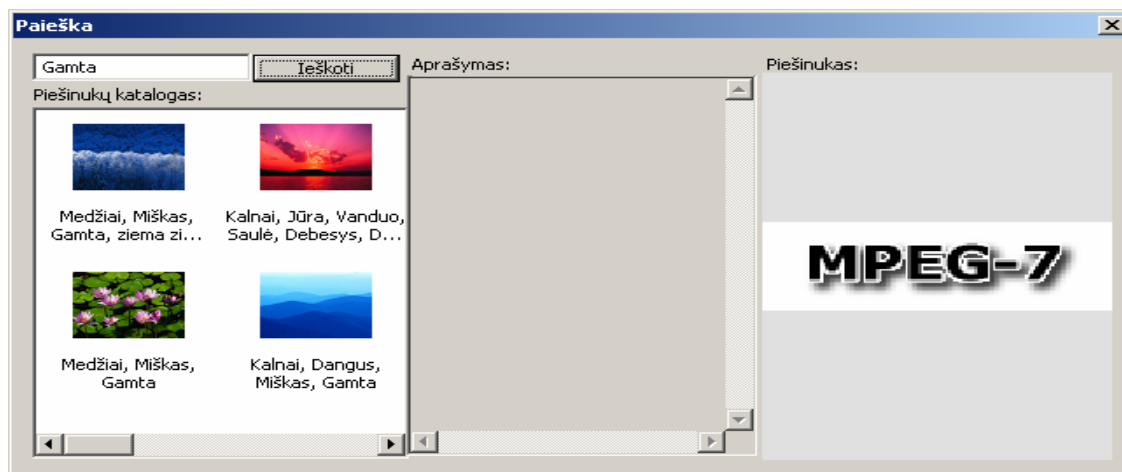
Meniu juosta sudaro trys pagrindines komandų grupes: Byla, Paieška ir Pagalba.

Byla- pagrindiniame meniu punkte galima iškelti anksčiau sukurtą žodyną, išsaugoti naujai sukurtą žodyną, įkelti paveiksluką, išsaugoti paveiksluką, išsaugoti sukurtą katalogą arba baigti darbą.

- **Skaityti žodyną** – nuskaityti ir iškelti anksčiau sukurtą žodyną.
- **Įrašyti žodyną** – naujai sukurtą žodyno išsaugojimas XML formate.
- **Pridėti paveiksluką** – įkelti i piešinių katalogą naują paveiksluką.
- **Įrašyti MPEG7** – išsaugoti informaciją apie piešinėlį Mpeg-7 formatu.

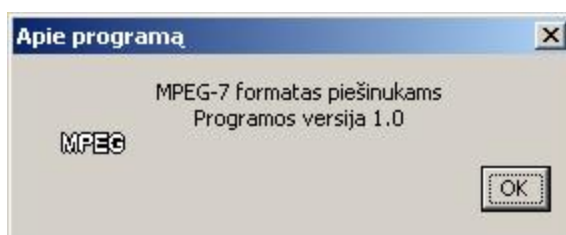
- **Įkelti MPEG-7** – įkelia ankščiau išsaugotą MPEG-7 failą .
- **Išsaugoti paveikslėlį** – išsaugo pasirinktą paveikslėlį į pageidaujama vietą.
- **Baigti darbą** – uždaro pagrindinį programos langą ir išjungia programą.

Meniu punktas **Paieška** turi vieną komandą - ieškoti teksto. Atsidarius naujam langui 16.pav. šalia komandinio mygtuko *ieškoti* įvedamas tekstas, pagal kurį mes ieškome paveikslėlio aprašo.



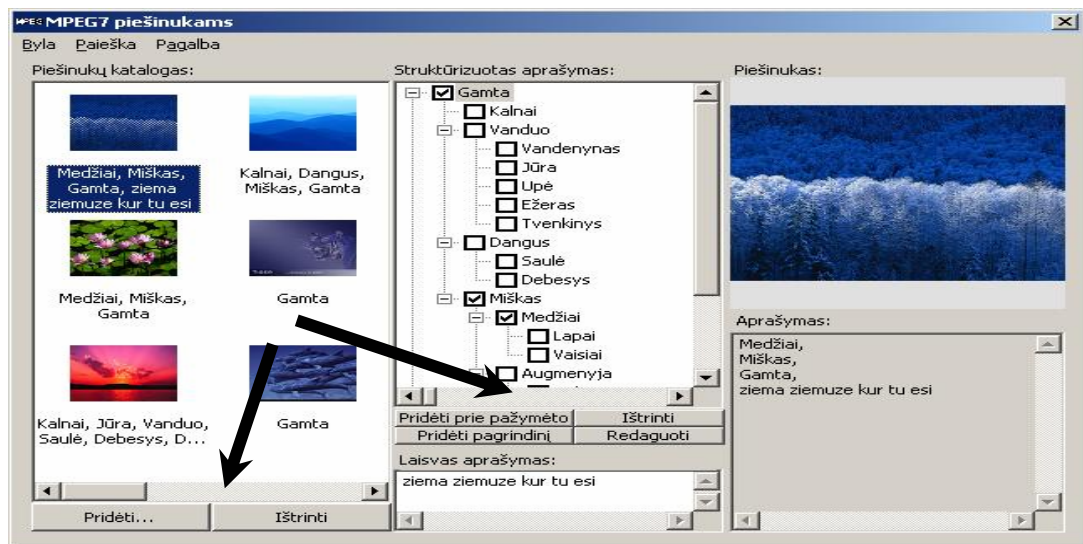
16.pav. Paieškos langas.

Meniu punktas pagalba iškviečia komanda Apie programą 17. pav.



17. pav. Apie programą

Patogesniam vartotojo darbui kiti komandiniai mygtukai išdėstyti programos lango viduryje ir kairiame apatiniame šone 18. pav.



18. pav. Pagrindinis langas

Šių komandų pagalba galima atlikti svarbiausius veiksmus dirbant su grafine medžiaga.

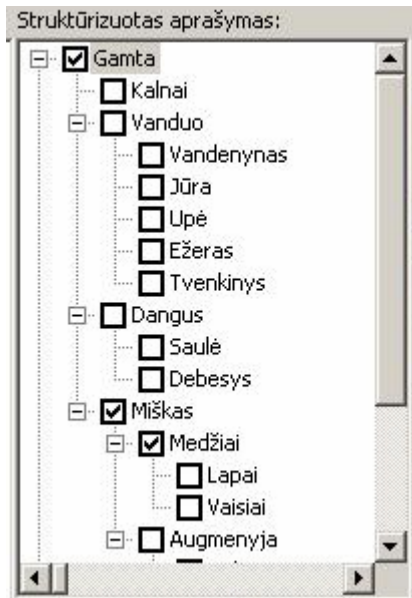
Programos langai

Pagrindinis programos langas išdalintas į šešis mažesnius langus. Į piešinių katalogo langą 19. pav. įkeliamas norimas aprašyti piešinukas.

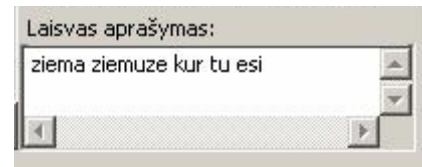


19. pav. Piešinių katalogas.

Paveikslėlio aprašymą galime vykdyti pasirinktinai struktūrizuoto aprašymo lange 20. pav. arba laisvo aprašymo lange 21. pav.



20. pav. Struktūrizuoto aprašymo langas.

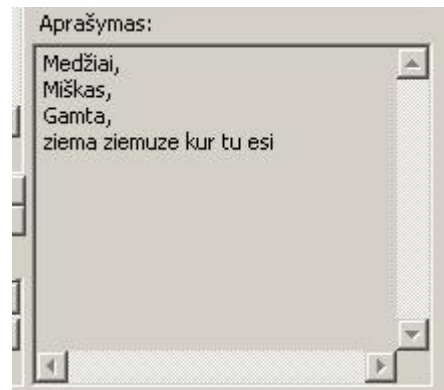


21. pav. Laisvo aprašymo langas

Lange „Piešinukas:“ 22. pav. Matomas aprašomo arba pasirinkto darbui piešinuko vaizdas. Po pasirinktuoju piešinuku lange Aprašymas matomas sukurtas paveikslėlio aprašas 23. pav.



22. pav. „Piešinukas:“



23. pav. Aprašymas:

Programos paieškos langą sudaro trys langai: „Piešinukų katalogas:“, „Aprašymas:“ ir „Piešinukas:“ langų savybės visiškai identiškios pagrindiniame programos lange esančių atitinkamų langų savybėms.

4.2 Programos MPEG instaliavimo dokumentas

4.2.1 Reikalavimai techninei įrangai

Reikalavimai sistemai:

- IBM PC 233 Mhz, 50MB HDD, 128 RAM, 16MB video, pelė, klaviatūra.
- Sistema veikia Windows 9X/N/2000/XP operacinėse sistemose.
- Būtinai sistemos komponentai GDI+ ir MSXML. Turi būti instaliuota gdiplus.dll ir bet kokia msxml versija.
- Rekomenduojamas spalvotas monitorius, kurio skiriamoji geba ne mažesnė nei 800x600.

4.2.2 Sistemos instaliavimas

Sistemos instaliuoti nereikia.

Galutinai sukompiliuotą sistemą sudaro vienas P.A.S.-mpeg7.exe failas. Norint naudotis sistema, tereikia perkelti jį į savo kompiuterį. Rekomenduojama į atskirą katalogą.

5. Testavimo specifikacija

5.1 Testavimo pasirinkimas

Bus vykdomas kelių tipų testavimas: automatizuota statinė analizė ir vizualinių požymių generavimo teisingumas.

5.2 Automatizuota statinė analizė

Šios analizės metu yra vykdomas išeities kodo teisingumo tikrinimas, jį atlieka kompiliatorius. Suradęs klaidingas programos išeities kodo vietas, kompiliatorius apie tai praneša, išvesdamas klaidingos kodo vietos eilutės numerį ir detalų klaidos aprašymą (kas sukėlė klaidingą situaciją).

Toks išeities kodo testavimo būdas tęsiasi tol, kol kompiliatorius neberanda nė vienos sintaksinės ar kitokio tipo klaidos (kintamojo tipo neatitikimas, kintamojo nepriskyrimas konkrečiam duomenų tipui ir kt.), tai leistų programai veikti be sutrikimų.

Ši analizė buvo vienas iš esminių sėkmingo kuriamos informacinės sistemos veikimo užtikrinimo būdų.

5.3 Vizualinių požymių generavimo teisingumas

Norėdami patikrinti mūsų algoritmų teisingumą, gautus rezultatus lyginsime su MPEG-7 pavyzdinės programinės įrangos (*reference software*) generuojamais tais pačiais požymiais.

Palyginus gautus rezultatus tarpusavyje ir nustatčius, kad aprašų rezultatai identiški, galime teigti, kad mūsų naudojami algoritmai atlieka teisingus skaičiavimus. Algoritmų teisingumą galima patikrinti ir interneto puslapyje [9].

5.4 Visų galimų būsenų testavimas

Šių bandymų atlikti nenumatoma, kadangi kuriama sistema nėra kritiškai svarbi laiko atžvilgiu, o įvykus kokiems nors sutrikimams, sistemą būtų galima atstatyti, paleidus ją iš naujo.

5.5 Testavimo medžiaga

1 etapas: remiantis MPEG7 formatą aprašančia XML-SCHEMA, kuriamas XMLinių failų formatas žodynui bei piešinukų katalogui. Testuojant tarpinius variantus, naudojamas puslapis <http://smil.nist.gov/M7Validation.html>, kuriame galima pasitikrinti MPEG7 failo teisingumą.

2 etapas: XML kodo generavimas su MSXML ir testavimas vėlgi tame pačiame puslapyje.

3 etapas: daromi ir testuojami smulkūs programos komponentai, kuriama programos infrastruktūra.

4 etapas: smulkūs komponentai jungiami į bendrą vartotojo sąsają, testuojamas jų veikimas kartu.

5 etapas: ieškoma resursų nutekėjimo, dinaminės atminties valdymo klaidų.

6 etapas: programa atiduodama testuoti vartotojams.

6. Rizikos įvertinimo ir mažinimo planas

Projekto rizikos:

Reikalavimų pasikeitimas

Programos pasikeitimas gali būti įtakotas programos kūrimo proceso. Kadangi pagrindinė idėja nustatyta, keistūsi tik atskiros detalės.

Su vartotoju susijusios rizikos:

PI atsisakymas

Vartotojas gali atsisakyti PI, tačiau projektas yra kuriamas ir moksliniais tikslais.

Sprendimas: Svarbiausia neapsiriboti tik vienu vartotoju, nes kuriamas produktas ganėtinai universalus. Jei vartotojas atsisako produkto dėl kainos, reikia atlikti su juo derybas, pasiūlyti pigesnes programinės įrangos aptarnavimo paslaugas ir pan., t. y. naudoti papildomas marketingo bei kainodaros priemones.

Naujų reikalavimų įvedimas bei senų keitimas

Į galimą sistemos praplėtimą atsižvelgiama projektavimo procese, kurio metu siekiama, kad sistema būtų kiek galima lankstesnė ir lengvai išplečiama. Papildomi reikalavimai pareikalautų laiko jiems įdiegti. Žinoma, vartotojas būtų informuotas apie tai, kad įdiegimams reikės papildomai laiko ir lėšų.

Proceso rizika:

Papildomų darbai

Papildomi darbai labai apsunkintų projekto baigimą laiku.

Sprendimas: Paaiškinti vartotojui apie galimą projekto uždelsimą, bandyti derėtis su juo, priešingu atveju, jam nesutikus, pirmiausiai dirbti prie svarbiausių projekto vietų, kad vartotojui atrodytų, jog praktiškai visas funkcionalumas egzistuoja ir projektas bus pristatytas laiku, likusį funkcionalumą įdiegiant vėliau.

Techninės rizikos:

Kompiuterių gedimas

Nors šių dienų kompiuterinė technika ganėtinai patikima, tačiau tokia tikimybė išlieka.

Sprendimas: Kompiuterio komponentų (išskyrus standųjį diską) gedimas labai didelės įtakos projekto eigai neturės, kadangi nėra sunku sugedusią komponentę pakeisti nauja, o projekto duomenims, pavyzdžiui, garso plokštės gedimas - nepakenktų.

Standžiojo disko gedimas

Tai labai įtakotų projekto eigą.

Sprendimas: Periodiškai susikurti projektui svarbios informacijos kopijas tiek į CD-R laikmenas, tiek ir į kitus standžiuosius diskus, tokiu atveju prarastos informacijos kiekis būtų nedidelis.

Darbo priemonių rizikos:

Darbuotojų komandos dydžio ir patirties rizikos:

Komandos dydis

Komanda pradinei projekto daliai atlikti yra pakankamo dydžio. Nors ją sudaro vienas narys, tačiau jo aplinkoje yra nemažai kolegų iš kurių, reikalui esant, jis gali gauti pagalbą. Projektui plečiantis komandoje turėtų būti daugiau narių.

Rizikas įvertiname 1-oje lentelėje:

1 lentelė. Rizikos faktoriai

Rizika	Tikimybė	Įtaka
Kompiuterių gedimas	0.2	3
Pavėluotas pristatymas	0.3	1
Reikalavimų pasikeitimai	0.02	3
Kompetencijos trūkumas	0.3	2
Kietojo disko gedimas	0.2	1
Vartotojo atmetimas	0.3	2
Papildomi darbai	0.5	1

Įtakos įverčiai:

1 – Labai blogai; 2 – Blogai; 3 – Pakenčiama

7. Išvados

1. Atlikus programinės įrangos ir MPEG-7 failo generavimo analizę, buvo nustatyti specializuotai paveikslėlių aprašymo programai keliami reikalavimai bei vartotojų poreikiai.
2. Sukurtos sistemos veikimui būtini du operacinės sistemos komponentai:
 - GDI+ bibliotekos;
 - MSXML parseris.
3. XML-SCHEMA puikiai tinka dokumentų aprašymui dėl savo lankstumo ir plečiamumo galimybių.
4. Sukurtos sistemos patrauklumas leidžia tikėtis, kad šis naujas standartas patenkins lūkeščius ir padės atsakyti į klausimus, kaip surasti, panaudoti ir valdyti šiuolaikinę vaizdo ir garso informaciją.

Literatūra

1. MPEG-7 Overview [interaktyvus]. [žiūrėta 2002-10-25], prieiga per internetą: <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>
2. International standard ISO/IEC 15938-1-8 Information technology-Multimedia content description interface First edition 2002-05-15
3. IBM MPEG-7 Annotation Tool [interaktyvus]. [žiūrėta 2002-12-15], prieiga per internetą: <http://www.alphaworks.ibm.com/tech/videoannex>
4. MPEG-7 Project at Columbia University [interaktyvus]. [žiūrėta 2002-12-15], prieiga per internetą: <http://www.ctr.columbia.edu/~ana/MPEG7/MPEG7project.html>
5. Canon Research Centre Europe Ltd [interaktyvus]. [žiūrėta 2002-12-15], prieiga per internetą: <http://www.cre.canon.co.uk/mpeg7ast/>
6. SINGINGFISH [interaktyvus]. [žiūrėta 2003-10-15], prieiga per internetą: <http://www.singingfish.com/>
7. 2.5.2 Стандарт MPEG-7 [interaktyvus]. [žiūrėta 2003-11-15], prieiga per internetą: http://citforum.parma.ru:8101/nets/semenov/2/25/mpeg_7.shtml
8. MPEG-7 eXperimentation Model (XM) [interaktyvus]. [žiūrėta 2002-11-15], prieiga per internetą: http://www.lis.ei.tum.de/research/bv/topics/mmdb/e_mpeg7.html
9. NIST MPEG-7 Validation Service [interaktyvus]. [žiūrėta 2004-5-10], prieiga per internetą: <http://smil.nist.gov/M7Validation.html>

8. Summary

The value of information often depends on how easy it can be found, retrieved, accessed, filtered and managed.

This challenging situation demands a timely solution to the problem. MPEG-7 is the answer to this need.

MPEG-7, formally named “Multimedia Content Description Interface”, is a standard for describing the multimedia content data that supports some degree of interpretation of the information’s meaning, which can be passed onto, or accessed by, a device or a computer code.

The main objective of this project is to create low complicated, small size, more concrete system, which main features are finding, retrieval, access, filtration and management of broad range of pictures.

9. Terminų ir santrumpų žodynas

2 lentelė. Terminų ir santrumpų žodynas

Pavadinimas	Paiškinimas
API	<i>Application Programming Interface</i> – programinės įrangos kūrimo priemonės.
MPEG-7	Moving Picture Experts Group – ISO/IEC 15938 standartas.
PĮ	Programinė įranga
Win API	<i>Windows Application Programming Interface</i> – Programinės įrangos kūrimo priemonės Windows aplinkoje.
XML	<i>eXtensible Markup Language</i> – Duomenų struktūrų standartas.