

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**PROGRAMŲ INŽINERIJOS KATEDRA**

Edvinas Mačiulis

**SINCHRONINIO MOKYMO PROCESO STANDARTIZAVIMAS**

**VAIZDO PASKAITŲ SISTEMOJE VIPS**

Magistro darbas

Vadovas: Prof. Aleksandras Targamadzė

**KAUNAS, 2008**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**PROGRAMŲ INŽINERIJOS KATEDRA**

Edvinas Mačiulis

**SINCHRONINIO MOKYMO PROCESO STANDARTIZAVIMAS**

**VAIZDO PASKAITŲ SISTEMOJE VIPS**

Magistro darbas

Recenzentas: .....

Dr. Dovilė Vojevodina

2008 05 26

Vadovas:.....

Prof. Aleksandras Targamadžė

2008 05 26

Autorius:.....

IFM-2/2 gr. stud. Edvinas Mačiulis

2008 05 26

**KAUNAS, 2008**

# TURINYS

<b>IVADAS</b>	<b>8</b>
<b>1. ĮRANKIO ĮGYVENDINIMO IR ESAMŲ SISTEMŲ ANALIZĖ</b>	<b>8</b>
1.1. ANALIZĖS TIKSLAS	8
1.2. TYRIMO SRITIS, OBJEKTAS IR PROBLEMA	9
1.3. ORGANIZACIJOS VEIKLOS ANALIZĖ	9
1.4. VARTOTOJŲ ANALIZĖ	10
1.4.1. Vartotojų aibė, tipai, savybės	11
1.4.2. Vartotojų tikslai ir problemos	11
1.5. PROBLEMOS SPRENDIMO METODŲ LITERATŪROS ŠALTINIuose ANALIZĖ	11
1.5.1. Panašių sistemų analizė	13
1.5.2. Programų sistemų kokybinis palyginimas	17
1.6. ARCHITEKTŪROS IR GALIMŲ ĮGYVENDINIMO PRIEMONIŲ VARIANTŲ ANALIZĖ	18
<b>2. LOKALUS VIDEO PASKAITŲ ĮRAŠYMO IR ĮRAŠŲ REDAGAVIMO ĮRANKIO VIPS SISTEMAI PROJEKTAS</b>	<b>19</b>
2.1. FUNKCINIAI REIKALAVIMAI	19
2.1.1. Veiklos sudėtis	19
2.1.2. Sistemos sudėtis	21
2.2. FUNKCINIAI REIKALAVIMAI IR REIKALAVIMAI DUOMENIMS	25
2.2.1. Funkciniai reikalavimai	25
2.2.2. Reikalavimai duomenims	25
2.3. NEFUNKCINIAI REIKALAVIMAI	26
2.3.1. Reikalavimai sistemos išvaizdai	26
2.3.2. Reikalavimai panaudojamumui	26
2.3.3. Reikalavimai vykdymo charakteristikoms	26
2.3.4. Reikalavimai veikimo sąlygoms	26
2.3.5. Reikalavimai sistemos priežiūrai	26
2.3.6. Reikalavimai saugumui	26
2.4. ARCHITEKTŪROS SPECIFIKACIJA	27
2.4.1. Apžvalga	27
2.4.2. Architektūros pateikimas	27
2.4.3. Architektūros tikslai	28
2.4.4. Sistemos statinis vaizdas	28
2.4.5. Sistemos dinaminis vaizdas	29
2.4.6. Išdėstymo vaizdas	34
2.4.7. Duomenų vaizdas	34
2.4.8. Kokybė	34
2.5. DETALIOS ARCHITEKTŪROS SPECIFIKACIJA	35
2.5.1. Kliento komponentas	35
2.6. TESTAVIMO MEDŽIAGA	39
2.6.1. Testavimo tikslai ir objektai	40
2.6.2. Testavimo apimtis	40
2.6.3. Testavimo planas	41
2.6.4. Testuojama programinė įranga	41
2.6.5. Sąsajos	41
2.6.6. Testavimo resursai	41
2.6.7. Testavimo įrankiai ir aplinka	42
2.6.8. Testavimo procedūra	42
<b>3. ĮRANKIO STANDARTIZAVIMO GALIMYBIŲ TYRIMAS</b>	<b>46</b>
3.1. EGZISTUOJANTYS STANDARTAI IR JŲ SAVYBIŲ ANALIZĖ	46
3.1.1. AICC	46
3.1.2. LOM	46
3.1.3. IMS LD	47
3.1.4. SCORM	48
3.2. TINKAMUMO IR KOKYBĖS KRITERIJŲ ĮVERTINIMAS	50
3.3. PASIRINKTO STANDARTO REALIZACIJOS REKOMENDACIJOS ĮRANKIUI	51
3.3.1. Turinio archyavimas	51

3.3.2.	<i>Mokymosi medžiagos pateikimo eiliškumo organizavimas .....</i>	<i>54</i>
<b>4.</b>	<b>IŠVADOS.....</b>	<b>54</b>
<b>5.</b>	<b>TERMINŲ SANTRUMPŲ ŽODYNAS .....</b>	<b>55</b>
<b>6.</b>	<b>LITERATŪRA.....</b>	<b>56</b>

# **STANDARDISATION OF SYNCHRONOUS LEARNING PROCESS IN VIDEO LECTURE AND PRESENTATION SYSTEM VIPS**

## **SUMMARY**

In this master thesis main topic is distance learning. It's another learning method more comfortable, but makes a little bit difficulties, when the intent is to deliver information and to make distance learning equal to the traditional. The main problem of distance learning is the presentation of the learning material. Lecturers and other people are trying to make learning process easier and more effective. It's characteristics depends on the tools, software and hardware, which is used to represent and deliver learning material. In this work we will know more about video presentation system ViPS and it's additional tool for local video lectures recording, when there is not internet. Presented information is about the value tools give us and requirements for it's functionality, architecture and results. Finally we will know more about distance learning standards, it's advantages and disadvantages and of course the possibility of implementation of the most suitable standard for the ViPS local recording tool.

## **LENTELĖS**

**1 lentelė.** Kuriamos ir esamų sistemų funkcijų palyginimas

**2 lentelė.** Veiklos įvykių sąrašas

**3 lentelė.** Funkcijos public Boolean doLogin(String uname, String upass) testavimas

**4 lentelė.** Funkcijos public List addSlides(path) testavimas

**5 lentelė.** Funkcijos public boolean changeSlidesPosition(int firstSlideIndex, int secondSlideIndex) testavimas

**6 lentelė.** Funkcijos public boolean changeSlideName(int index, String name) testavimas

**7 lentelė.** Funkcijos public boolean getRecord(int recordID) testavimas

**8 lentelė.** Funkcijos public boolean changeTime(int slideIndex, SimpleDateFormat time) testavimas

**9 lentelė.** Funkcijos public boolean sendRecord() testavimas

## **PAVEIKSLĖLIAI**

**1 paveikslėlis.** Bendravimo ir bendradarbiavimo priemonės pagal sąveikos tipą

**2 paveikslėlis.** Centra® Live bendravimo ir bendradarbiavimo aplinka

**3 paveikslėlis.** Horizon Wimba bendravimo ir bendradarbiavimo aplinka

**4 paveikslėlis.** Interwise bendravimo ir bendradarbiavimo aplinka

**5 paveikslėlis.** Veiklos konteksto schema

**6 paveikslėlis.** Panaudojimo atviejų sąrašas

**7 paveikslėlis.** Duomenų schema

**8 paveikslėlis.** Sistemos paketas

**9 paveikslėlis.** Klasių diagrama

**10 paveikslėlis.** Įrašo kontroliavimo bendradarbiavimo diagramos

**11 paveikslėlis.** Skaidrių valdymo bendradarbiavimo diagramos

**12 paveikslėlis.** Įrašo eksportavimo/importavimo bendradarbiavimo diagramos

**13 paveikslėlis.** Įrašo skaidrės laiko keitimo bendradarbiavimo diagrama

- 14 paveikslėlis.** Sekų diagrama
- 15 paveikslėlis.** Įrašų valdymo veiklos diagrama
- 16 paveikslėlis.** Skaidrių valdymo veiklos diagrama
- 17 paveikslėlis.** Importavimo/Eksportavimo veiklos diagramos
- 18 paveikslėlis.** Išdėstymo vaisdas
- 19 paveikslėlis.** Paketo klasių diagrama
- 20 paveikslėlis.** Mokymosi turinio skaidymas ir grupavimas
- 21 paveikslėlis.** Turinio struktūros pavyzdys

## **IVADAS**

Žmonėms, kuriems informacinės technologijos yra neatsiejama gyvenimo dalis, gerai žinoma jog IT pasaulyje kas buvo nauja šiandien, rytoj gali būti jau visiems gerai žinoma, o vėliau dažniausiai ir pasenę. Kas dieną skaitydami naujienas sužinome apie sukurta naują programinę įrangą, kuri padeda mums spręsti šių dienų problemas ar tiesiog palengvina darbą, mokymasi, laisvalaikio organizavimą ar tiesiog leidžia namie būnant tvarkyti asmeninius rūpesčius.

Distancinis mokymas nėra labai senas mokymosi ir mokymo būdas, jį labai stipriai įtakoja naujų technologijų kūrimas, ypač informacinių, telekomunikacinių technologijų vystymasis. Galime pastebėti, kad nuotolinio mokymosi įrankių vystymosi progresas yra pakankamai greitas, ir tai gerina studijų kokybę bei patogumą. Nauji įrankiai užtikrina greitesnį ir efektyvesnį mokymosi medžiagos rengimą, o taip pat ir geresnį informacijos įsisavinimą, kadangi kuriant juos remiamasi jau turima savo ir kitų kūrėjų patirtimi.

Šiuo metu Lietuvoje pagrindiniai naudojami įrankiai:

- WebCT
- Moodle
- ViPS
- CDK
- WebReg

ViPS – įrankis skirtas transliuoti internetu realiu laiku vykstančias transliacijas, konferencijas, o taip pat įrašytus anksčiau vykusius įrašus. Sistema pasiekama internetu, naudojantis naršykle, prisijungus galima stebėti transliacijas ar įrašus, o taip pat įrašinėti video presentacijas. Tačiau kartais nutinka taip, kad dėl techninių kliūčių dingsta interneto ryšys ir norint atlikti video presentacijos įrašymą susiduriama su pagrindine problema – naršyklė mums neatverčia tinklapio kurį naudodami galėtume alikti įrašą.

## **1. ĮRANKIO ĮGYVENDINIMO IR ESAMŲ SISTEMŲ ANALIZĖ**

### **1.1. Analizės tikslas**

Analizė svarbus programinės įrangos kūrimo etapas, kuo tiksliau ir efektyviau ji atliekama, tuo geresnis ir aiškesnis tampa pačios sistemos, projektavimo, bei kūrimo procesas. Šiuo atveju kuriama nuotolinio mokymosi sistema, taigi esminiai tikslai yra suprasti su kokiomis problemomis susiduria nuotoliniu būdu besimokantys žmonės, o taip pat ir įsisavinti teikiančiųjų, rengiančių metodinę medžiagą kasdieninius sunkumus. Be



abejo išskylančios kliūtys taip pat priklauso ir nuo organizacijos veiklos, darbuotojų bei vartotojų.

Analizės metu svarbu suprasti kliūtis, aplinkos įtaką būsimai sistemai. Šio etapo metu siekiama išsiaiškinti problemų sprendimo būdus literatūros šaltiniuose. Tai apsaugo nuo tų pačių klaidų kartojimo, o taip pat suteikia galimybę pažvelgti giliau ir prieti prie originalesnių, gudresnių ir patogesnių sprendimo variantų. Susipažinimas su panašiomis sistemomis taip pat svarbus, kad rezultate nebūtų gaunamas labai panašus ar identiškas variantas, nes kartais pirkti PĮ pigiau nei kurti savo iš naujo, dažniausiai taip ir būna. Galiausiai, siekiant geriausio rezultato prieš priimant sprendimą reikia išanalizuoti galimas kūrimo priemones, bei esamus architektūros variantus. Tokiu būdu įmanoma išspręsti problemas, kurios galėtų kilti ateityje, o taip pat iš esmės pakeisti kūrimo proceso spartą ar PĮ kokybę.

Galutinis analizės rezultatas yra siekiamos sukurti sistemos aiškus apibrėžimas.

## **1.2. Tyrimo sritis, objektas ir problema**

**Tyrimo sritis** – nuotolinis mokymasis, jo proceso tobulinimas bei mokymosi medžiagos gerinimas, naudojant įrankius, skirtus naudoti lokaliame kompiuteryje.

**Objektas** – sinchroninio mokymosi video paskaitų sistema ViPS.

**Problema.** ViPS – įrankis skirtas transliuoti internetu realiu laiku vykstančias transliacijas, konferencijas, o taip pat įrašytus anksčiau vykusius įrašus. Įrankis pasiekiamas internetu, naudojantis naršykle, prisijungus galima stebėti transliacijas ar įrašus, o taip pat įrašinėti video presentacijas. Tačiau kartais nutinka taip, kad dėl techninių kliūčių dingsta interneto ryšys ir norint atlikti video presentacijos įrašymą susiduriama su pagrindine ir esmine problema – neįmanoma atlikti įrašo, kadangi interneto ryšys būtinas.

## **1.3. Organizacijos veiklos analizė**

**Distancinio mokymo centras** - tai universiteto padalinys, įkurtas 1996 m. Kauno technologijos universitete, dalyvaujant PHARE finansuojamame projekte „Daugiašalis bendradarbiavimas distancinio švietimo srityje (Multi-Country Co-Operation in Distance Education)“.

**Misija** - kurti ir plėtoti informacijos ir komunikacijos technologijomis paremtą aukštojo mokslo studijų ir tęstinio mokymosi sistemą.

**Veiklos tikslai:**

- Moderniomis telekomunikacijų, informacijos ir vaizdo konferencijų technologijomis paremtos aukštojo mokslo studijos
- Aukštųjų mokyklų pedagogų ir studentų kompetencijos plėtotė, vystant skaitmeninei komunikacijai ir veiklai būtinus gebėjimus
- Aukštųjų mokyklų partnerystė vystant distancinį mokymąsi Lietuvoje
- Suaugusiųjų (profesinio, tęstinio) mokymosi galimybių plėtojimas
- Narystė pasaulio tęstinio mokymosi organizacijų tinkluose

#### **Pagrindinės veiklos kryptys:**

- Metodinės medžiagos rengimas DM kursų autoriams, kuratoriams, administratoriams
- Dėstytojų ir kuratorių mokymas
- DM kursų kūrimas ir teikimas
- DM bibliotekos komplektavimas
- Distancinio mokymosi tyrimai (DM poreikių, metodikų, technologijų ir kt. srityse)
- Vaizdo konferencijų tinklo koordinavimas
- Virtualių mokymosi terpių kūrimas ir palaikymas
- Garso ir vaizdo mokymosi priemonių kūrimas
- Bendradarbiavimas su pramonės ir verslo sektoriumi, formalaus ir neformalaus švietimo organizacijomis, vyriausybėmis bei nevyriausybėmis struktūromis
- Bendradarbiavimas tarptautiniuose projektuose

**Narystė organizacijose** - Distancinio mokymo centras yra Nacionalinės distancinio mokymo asociacijos (NDMA), Lietuvos suaugusiųjų švietimo asociacijos (LSŠA), Baltijos regiono distancinio mokymo asociacijos tinklo (BOLDIC), Europos distancinio ir e.mokymosi tinklo (EDEN) narys.

#### **1.4. Vartotojų analizė**

Šioje skiltyje išanalizuosim ViPS sistemoje egzistuojančius vartotojų tipus, bei tipus ir savybes.

#### 1.4.1. Vartotojų aibė, tipai, savybės

- Vartotojų aibė video paskaitų sistemoje VIPS – tai asmenys, besimokantys nuotoliniu būdu arba teikiantys nuotolinio mokymosi paslaugas.
- Pagal tai skirstomi ir pagrindiniai vartotojų tipai:
  - Dėstytojai – kuria kursus, susideda paskaitų skaidres, praveda paskaitas.
  - Studentai – dalyvauja paskaitoje.
  - Kiti galimi sistemos vartotojai:
    - Administratoriai – sistemos prižiūrėtojai, turintys visas teises.
    - Asistentai – tai yra dėstytojų pagalbininkai, apribotas dėstytojo vartotojo variantas.
    - Svečiai – yra dar labiau apribotų teisių studento tipo vartotojų variantas.

#### 1.4.2. Vartotojų tikslai ir problemos

- Studento tikslas yra rasti reikiamą medžiagą ir ją įsisavinti.
- Dėstytojo tikslas yra pateikti medžiagą, prvesti paskaitą.
- Asistento tikslas yra pavaduoti dėstytoją techniniais klausimais.
- Administratoriaus pagrindinė veikla ir atsakomybė – stabilus sistemos veikimas, komunikacija su vartotojais ir pagalba jiems.

#### 1.5. Problemos sprendimo metodų literatūros šaltiniuose analizė

Bendravimo ir bendradarbiavimo aplinka – tai integruotų ir tarpusavyje sąveikaujančių bendravimo ir bendradarbiavimo priemonių visuma. Bendravimo ir bendradarbiavimo priemonės pagal sąveikos laiką galime suskirstyti į dvi grupes: sinchroninės (veiksmai atliekami tuo pačiu metu) ir asinchroninės priemonės (veiksmai atliekami nebūtinai tuo pačiu metu) (1 pav.).



1 paveikslėlis. Bendravimo ir bendradarbiavimo priemonės pagal sąveikos tipą

**Sinchroninės bendravimo priemonės:**

- tekstiniai pokalbiai,
- pokalbiai balsu,
- vaizdo ryšys.

**Sinchroninės bendradarbiavimo priemonės:**

- pateiktys,
- naršymas drauge,
- bendras programos lango valdymas,
- bendros diagramų kūrimo priemonės,
- žaidimai tinkle.

**Asinchroninės bendravimo priemonės:**

- el. paštas,
- diskusijų forumai,
- naujienos.

**Asinchroninės bendradarbiavimo priemonės:**

- bendras teksto redaktorius (pvz., Wiki),
- failų keitimosi priemonės,
- srautinė multimedija,
- grupių ir teisių valdymas.

Bendrinės bendravimo ir bendradarbiavimo priemonės dažniausiai veikia kaip savarankiškos informacinės sistemos. Bendravimo ir bendradarbiavimo priemonės – viena iš virtualios mokymosi aplinkos dalių, todėl svarbi jų integracija ir kontekstinė sąveika su virtualia mokymosi aplinka ir kitomis e. mokymosi informacinėmis sistemomis. Dėl šių priežasčių analizėje nagrinėjame tik tas bendravimo ir bendradarbiavimo priemones, kurios yra pritaikytos e. mokymosi poreikiams ir integruojamos į virtualią mokymosi aplinką.

Šios analizės tikslas: apžvelgti labiausiai paplitusias bendravimo ir bendradarbiavimo aplinkas, išryškinti jų vystymosi tendencijas ir naudojimo modelius, apibrėžti tolimesnes ViPS aplinkos vystymosi gaires.

### **1.5.1. Panašių sistemų analizė**

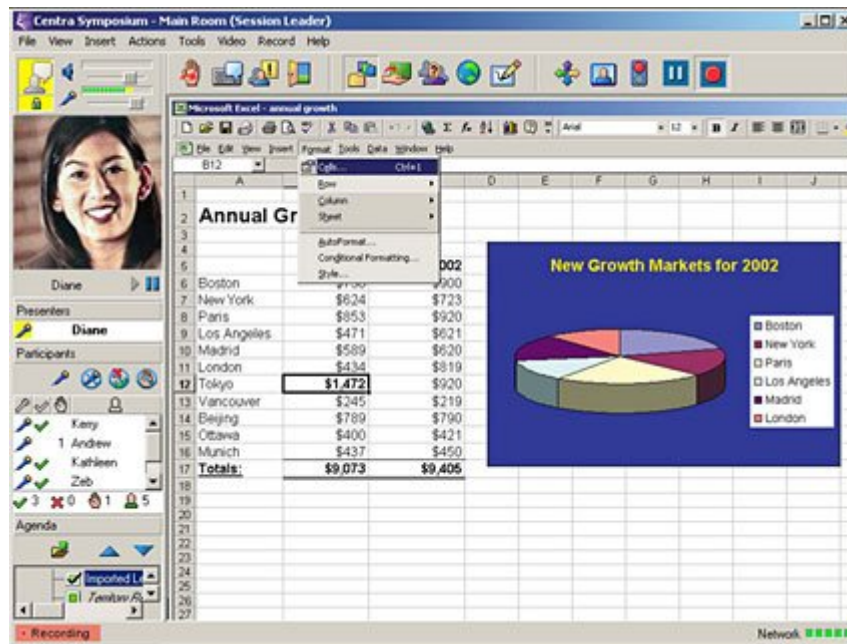
Šioje skiltyje pateikiamos panašios sistemos ir jų savybės.

#### **1.5.1.1. Centra®**

Centra® įrankis suteikia galimybę virtualiai mokytis, kuris išplečia Saba kompanijos sukurtus mokymosi valdymo sprendimus. Sistema sudaryta iš:

1. Centra Live™ for Virtual Classes – bendravimas grupėmis realiu laiku iš skirtingų vietų pasaulyje
2. Virtual Classes for Universities and Schools – studentų ir dėstytojų bendravimo interaktyvi aplinka, suteikianti tokias pačias galimybes kaip ir bendraujant realioje auditorijoje.
3. Centra Live™ for Web Seminars – efektyvus, greitas ir nebrangus būdas komunikuoti didelėms auditorijoms
4. Centra Live™ for eMeetings – galimybė virtualiai organizuoti žmonių pasitarimus
5. Centra Knowledge Center™ - informacijos ir resursų valdymas iš vienos centralizuotos vietos.
6. Third Party Integration – galimybė integruoti aplikacijas, technologijas bei darbalaukio įrankius kurie jau egzistuoja organizacijoje.

Kaip galime suprasti iš pateikto priemonių sąrašo, Centra® suteikia daug galimybių ne tik mokymosi institucijoms, kas yra daug žadantis veiksnys integruojant ateities e.komercijos ir e.mokymosi sistemas į vientisą decentralizuotą sąveikaujančių tinklo paslaugų sistemą. Paveikslėlyje matome Web Seminars sistemos modeliavimo dalis (2 pav.)



**2 paveikslėlis.** Centra® Live bendravimo ir bendradarbiavimo aplinka.

Paimta iš [http://www.ivci.com/images/centra\\_symposium.jpg](http://www.ivci.com/images/centra_symposium.jpg)

Realiu laiku prisijungę dalyviai gali:

- dalyvauti apklausose
- kelti rankas
- kvatotis
- ploti
- naršymas drauge
- susirašinėti visiems matant ir privačiai
- naudoti baltą lentą grupėmis
- galimybė vadovui leisti bendrauti studentams tarpusavyje

Įrankis leidžia ne tik realiu laiku mokytis, bet ir prisijungus peržiūrėti įrašytas sesijas. Sistemoje taip per realizuotas auditorijos valdymas:

- studentų skirstymas į grupes
- studentų aktyvumo kontroliavimas
- leidžiama destytojams iš bet kurios norimos vietos kontrliuoti sesijas

- neverbalinis gryžtamasis ryšys: taip/ne, žingsnis atgal, rankos kėlimas

Turinyje gali būti tekstas, paveikslėliai(įskaitant GIF formatą), taip pat galima integruoti flash, javascript, audio, video įrašą. MS PowerPoint skaidrės gali būti realiu laiku įkeliamos, animacija ir įterpti multimedijos objektai išsaugomi. Žinių vertinimas atliekamas:

- realiu laiku apklausiant ar studentai supranta medžiagą
- formalūs vertinimo įrankiai su vertinimo balais
- individualių ir bendrų atsakimų ataskaitos

Taip pat sistema palaiko daugiakalbiškumą.

### **1.5.1.2. Horizon Wimba**

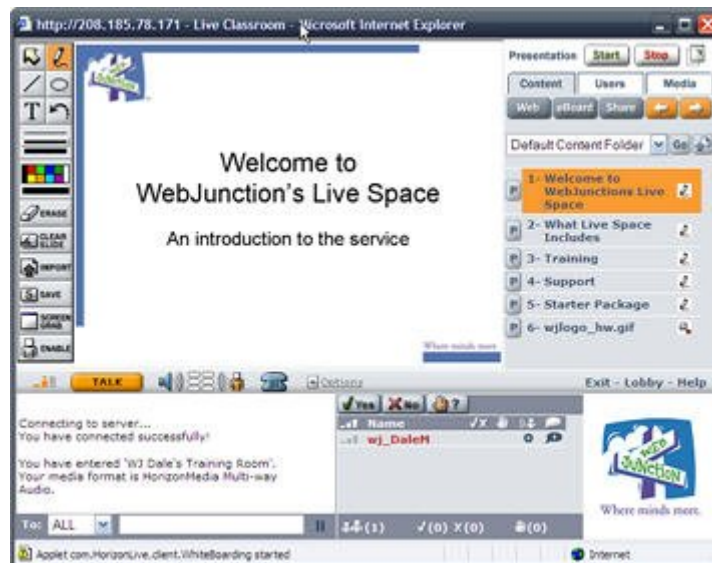
Horizon Wimba aplinka, panaši į Centra aplinką, daugiausiai skirta bendravimui ir bendradarbiavimui virtualioje klasėje.

Įrankis sudarytas iš dalių:

- Communication Bundle, jį sudaro produktai:
  - Live Classroom
  - Pronto
  - Wimba Voice Tools
- Content Bundle, kurį sudaro šie produktai:
  - Course Genie
  - Diploma
- Course Management System Integrations

Horizon Wimba aplinka turi panašias priemones kaip ir Centra aplinka: bendras kompiuterio programų valdymas (angl. application sharing), apklausos (taip/ne, kelių variantų atsakymas į klausimą), puslapių naršymas drauge, baltoji lenta (bendradarbiavimas piešiant), vaizdo įrašo valdymas, pokalbiai (privatūs ir bendri), multimedija medžiagos pateiktys (Flash ir Shockwave filmukai, JavaScript, animuoti paveikslėliai, srautinis garsas ir vaizdas). Horizon Wimba neturi grupių valdymo priemonės, tačiau naudoja pažangias garso perdavimo technologijas, todėl ši aplinka veikia net ir per lėtą interneto ryšį (56kbps).

Aplinka naudoja tik interneto naršyklę su Java. Horizon Wimba yra specializuota švietimo institucijų poreikiams. Žemiau pateikiama sistemos aplinka, kur matyti skaidrių modeliavimo dalis, sistemos komunikacijos su serveriu monitoringas. (3 pav.)



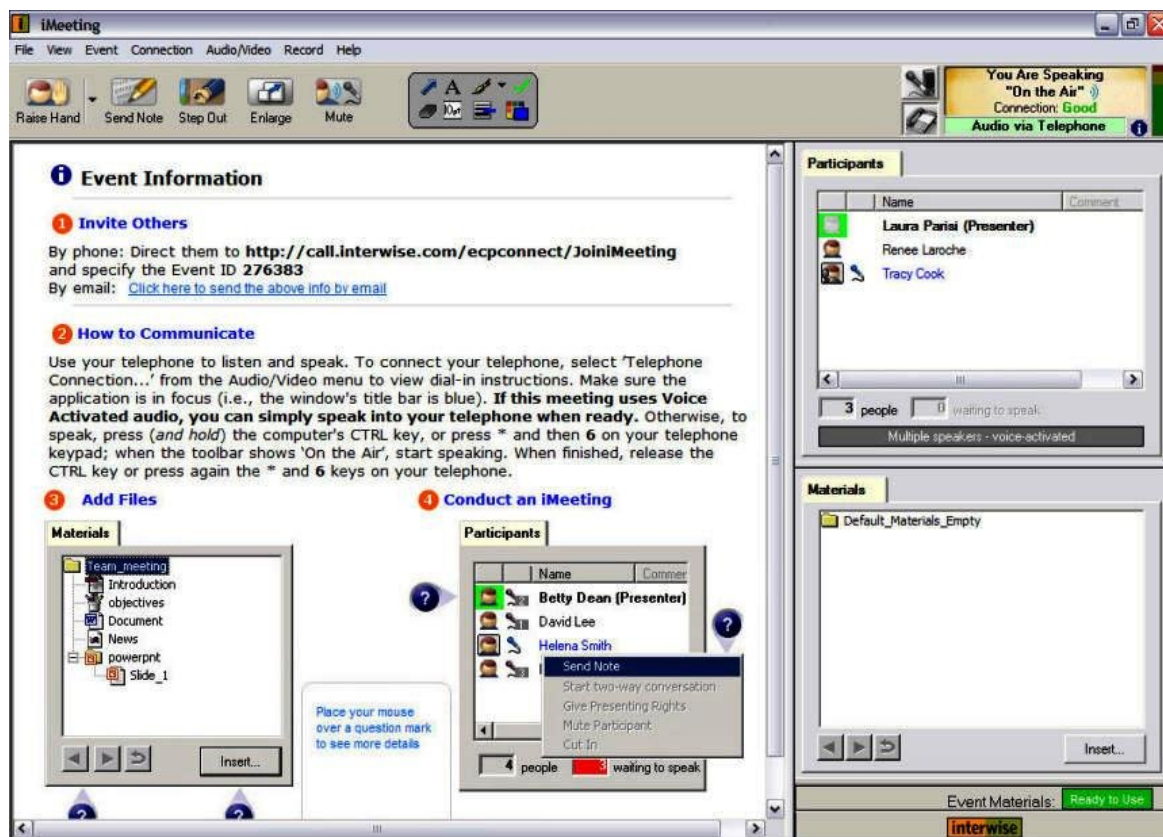
**3 paveikslėlis.** Horizon Wimba bendravimo ir bendradarbiavimo aplinka  
Paimta iš [http://static.flickr.com/139/323426631\\_c810ce1ef6\\_o.jpg](http://static.flickr.com/139/323426631_c810ce1ef6_o.jpg)

### 1.5.1.3. Interwise Connect™

Interwise Connect™ Sistemos galimybės:

- Konferencijos balsu
- Posėdžiai (meetings) internetu
- Virtualus mokymasis
- Globalūs darbuotojų susitikimai naudojant vidinį tinklą arba internetą
- Konferencijų įrašymai





4 paveikslėlis. Interwise bendravimo ir bendradarbiavimo aplinka

Paimta iš <http://www.ap-verlag.de/Online-Artikel/Online1.JPG>

*Interwise* sistemoje numatyti skirtingi sąveikos būdai: individualaus konsultavimo (angl. *Mentoring*), susitikimų, virtualios klasės, seminaro, transliacijos. *Interwise* aplinkoje yra šios priemonės(4 pav.): bendras kompiuterio programų valdymas (angl. *application sharing*), apklausos (taip/ne, pasirinkti atsakymą į klausimą iš kelių variantų), naršymas drauge, baltoji lenta (bendradarbiavimas piešiant), grupių kūrimas ir valdymas, vaizdo įrašo valdymas, pokalbiai (privatūs ir bendri), multimedija medžiagos pateiktys (Flash ir Shockwave filmukai, JavaScript, animuoti paveikslėliai, srautinis garsas ir vaizdas). *Interwise* aplinka reikalauja įdiegti specialią programinę įrangą ir veikia tik naudojant Internet Explorer naršyklę. Aplinka veikia tik Microsoft Windows aplinkoje.

### 1.5.2. Programų sistemų kokybinis palyginimas

Kadangi KTU distancinio mokymo centras turi savo sukurta sistemą, deja nepavyko rasti įrankių kurie atliktų mums reikiamas funkcijas.(1 lentelė)

1 lentelė. Kuriamos ir esamų sistemų funkcijų palyginimas

	Naujos ViPS galimybės	Centra	Horizon Wimba	Interwise

Įrašų importavimas iš sistemos	+	-	-	-
Įrašų eksportavimas į sistemos	+	-	-	-
Įrašinėjimas be interneto ryšio	+	-	-	-
Lokalus įrašo redagavimas	+	-	-	-
Įrašų konvertavimas pagrindinei sistemai tinkamu formatu	+	-	-	-

## 1.6. Architektūros ir galimų įgyvendinimo priemonių variantų analizė

Sistemos paskirtis ir veikimo principas iš anksto apibrėžia sistemos naudojimo būdą. Peržiūrėjus įrankio reikalavimus matome, jog tai bus programinė įranga eksploatuojama stacionariame arba nešiojamame kompiuteryje, iš kurio demonstruojama mokomoji medžiaga.

Realizacijos galimybių bei variantų iš tikro labai daug, kūrimo aplinkų su laiku tik daugėja, taigi turime tokius pagrindinius pasirinkimo variantus:

- Microsoft .NET platforma – C#
- Sun microsystems – JAVA
- C++
- Delphi

Kaip matyti turime didelį pasirinkimą, tačiau kiekviena platforma turi savo pliusų bei minusų.

**C# programavimo kalba.** Kaip ir Microsoft programinė įranga, ši kalba plačiai paplitusi. Savybės:

- .NET oficiali realizacija yra tik MS Windows aplinkai
- .NET realizacija trečių šalių nėra patikima, bei perkeltant į kitą sistemą reikia papildomų bibliotekų
- Sistemos veikia panašiai kaip ir Java kalba kurta PĮ.

**JAVA.** Labai paplitusi programavimo kalba, ja kuriamos tiek Web aplikacijos tiek desktopinės programos, bei PĮ mobiliems telefonams. Turi galingą aplikacijų programavimo sąsają, kurios realizuoja daug įvairių Java Enterprise edition programavimo technologijų, gerokai palenginančių kūrimo procesą. Savybės:

- JVM – java virtuali mašina realizuota daugelyje pagrindinėse ir dažniausiai sutinkamose operacinėse sistemose, sukurtos sistemos portabilios, veikia beveik visur.

- Programinės įrangos veikimas reikalauja daugiau resursu nei iš anksto sukompiliuotos sistemos konkrečiai OS.

**C++.** Senesnė nei anksčiau minėtos programavimo kalbos, tačiau išlaikanti savo poziciją programavimo rinkoje. Be jos neapseinama tiek kuriant programas MS Windows aplinkai tiek Unix tipo operacinėms sistemoms, o taip pat kai kurios mobiliųjų telefonų OS parašytos būtent šia programavimo kalba. Savybės:

- PĮ kompičiuojama konkrečiai OS.
- Greitesnis, nei prieš tai minėtų, sukompiliuotų sistemų veikimas.

**Delphi.** Kitaip dar vadinamas objektinis paskalis, kadangi sintaksė beveik nesiskirtia, kadangi kalba objektinė tai yra ir sintaksės papildymų. Savybės:

- PĮ kompičiuojama konkrečiai OS.
- Greitesnis, nei prieš tai minėtų, sukompiliuotų sistemų veikimas.

Pagrindiniai reikalavimai įrankiui, kurie įtakoja pasirinkimą konkretios programavimo kalbos yra:

- Paprastas sistemos perkeliamumas
- Sklandus įrankio funkcionavimas
- Nedideli reikalavimai kompiuterio techninei įrangai

Apžvelgus galimus variantus matome, kad tinkamiausia mums yra Java programavimo kalba.

## **2. LOKALUS VIDEO PASKAITŲ ĮRAŠYMO IR ĮRAŠŲ REDAGAVIMO ĮRANKIO VIPS SISTEMAI PROJEKTAS**

### **2.1. Funkciniai reikalavimai**

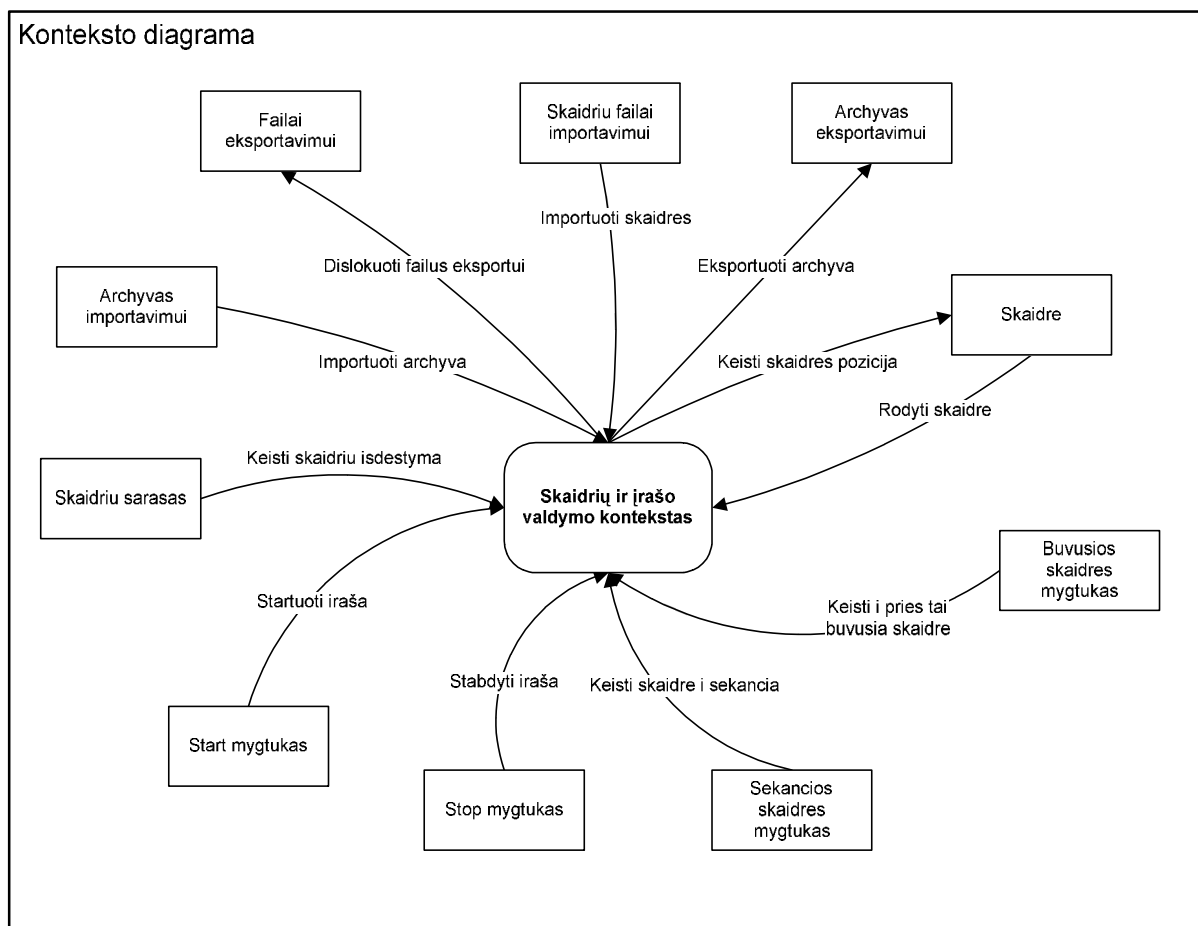
Šioje skiltyje pateikiami sistemos funkciniai reikalavimai, o taip pat su jais susijusi informacija: veiklos sudėtis, sistemos sudėtis.

#### **2.1.1. Veiklos sudėtis**

Veiklos sudėties aiškesniam supratimui pateiksime informaciją apie veiklos kontekstą ir veiklos padalinimą.

##### **2.1.1.1. Veiklos kontekstas**

Žemiau pateikiama veiklos konteksto schema, kurioje matyti tiek įeinantys tiek išeinantys duomenys.(5 pav.)



5 paveikslėlis. Veiklos konteksto schema

### 2.1.1.2. Veiklos padalinimas

Veiklos padalinime susiejami įvykiai su įeinančiais ir išeinančiais duomenimis.(2 lentelė)

2 lentelė. Veiklos įvykių sąrašas

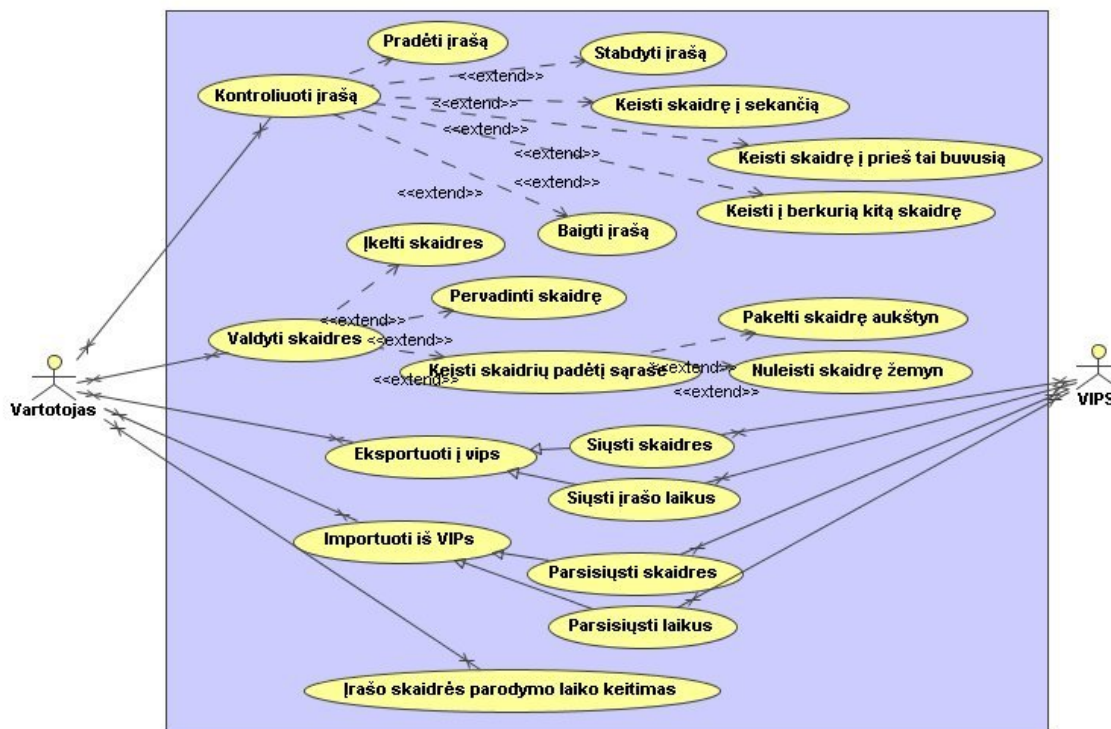
Nr.	Eil.	Įvykio pavadinimas	Įeinantys/Išeinantys informacijos srautai
1		Įrašo startavimas	Startavimo laikas
2		Įrašo stabdymas	Stabdymo laikas
3		Skaidrės keitimas sekačia	Skaidrės duomenys
4		Skaidrės keitimas prieš tai buvusią	Skaidrės duomenys
5		Skaidrės parodymas	Skaidrės duomenys
6		Skaidrės pozicijos keitimas	Skaidrės pozicija
7		Įrašo archyvo eksportavimas	Įrašo duomenys ir skaidrės
8		Skaidrių importavimas	Skaidrių vieta failų sistemoje
9		Failų eksportui dislokacija	Skaidrių failai
10		Archyvo importavimas	Skaidrių archyvas
11		Skaidrių išdėstymo keitimas	Skaidrių sąrašas

## 2.1.2. Sistemos sudėtis

Sistemos sudėtis aprašoma panaudojimo atviejų diagrama ir panaudojimo atviejų detalesniu aprašymu.

### 2.1.2.1. Sistemos ribos

Žemiau pateiktame paveikslėlyje matyti panaudojimo atviejai bei aktoriai. (6 pav.)



6 paveikslėlis. Panaudojimo atviejų sąrašas

### 2.1.2.2. Panaudojimo atviejų sąrašas

Šioje skiltyje pateikiami auksčiau paveikslėlyje matyti panaudojimo atviejai, pateikiant papildomos informacijos apie juos.

1. PANAUDOJIMO ATVEJIS: Įrašo kontroliavimas -> Įrašo pradėjimas

**Aktoriai:** Vartotojas

**Aprašas :** Pradedama įrašą

**Prieš-sąlygos:** skaidrių įkėlimas

**Sužadinimo sąlyga:** įrašo pradėjimo mygtuko spragtelėjimas

**Po sąlyga:** prasidėjo laiko skaičiavimas

2. PANAUDOJIMO ATVEJIS: Įrašo kontroliavimas -> Skaidrės keitimas į sekančią

**Aktoriai:** Vartotojas

**Aprašas:** Pakeičia skaidrę

**Prieš-sąlygos:** skaidrių įkėlimas

**Sužadinimo sąlyga:** sekančios skaidrės mygtuko spragtelėjimas

**Po sąlyga:** rodoma kita skaidrė

3. PANAUDOJIMO ATVEJIS: Įrašo kontroliavimas -> Skaidrės keitimas į prieš tai buvusią

**Aktoriai:** *Vartotojas*

**Aprašas:** Pakeičia skaidrę

**Prieš-sąlygos:** skaidrių įkėlimas

**Sužadinimo sąlyga:** prieš tai buvusios skaidrės mygtuko spragtelėjimas

**Po sąlyga:** rodoma kita skaidrė

4. PANAUDOJIMO ATVEJIS: Įrašo kontroliavimas -> Skaidrės keitimas į bet kurią kitą

**Aktoriai:** *Vartotojas*

**Aprašas:** Pakeičia skaidrę

**Prieš-sąlygos:** skaidrių įkėlimas

**Sužadinimo sąlyga:** skaidrės pasirinkimas pele iš sąrašo

**Po sąlyga:** rodoma kita skaidrė

5. PANAUDOJIMO ATVEJIS: Įrašo kontroliavimas -> Įrašo stabdymas

**Aktoriai:** *Vartotojas*

**Aprašas:** Stabdo įrašą

**Prieš-sąlygos:** pradėtas įrašas

**Sužadinimo sąlyga:** įrašo stabdymo mygtuko parinkimo spragtelėjimas

**Po sąlyga:** sustabdytas įrašas

6. PANAUDOJIMO ATVEJIS: Įrašo kontroliavimas -> Įrašo baigimas

**Aktoriai:** *Vartotojas*

**Aprašas:** Baigia įrašą

**Prieš-sąlygos:** pradėtas įrašas

**Sužadinimo sąlyga:** įrašo baigimo mygtuko parinkimo spragtelėjimas

**Po sąlyga:** baigtas įrašas, paruoštas eksportavimui, saugojimui

7. PANAUDOJIMO ATVEJIS: Skaidrių valdymas -> Skaidrių įkėlimas -> Skaidrių įkėlimas iš katalogo

**Aktoriai:** *Vartotojas*

**Aprašas:** Įkelia skaidres

**Prieš-sąlygos:** nėra

**Sužadinimo sąlyga:** įkėlimo iš katalogo mygtuko parinkimo spragtelėjimas

**Po sąlyga:** skaidrių sąrašas paruoštas redagavimui, įrašo pradėjimui

8. PANAUDOJIMO ATVEJIS: Skaidrių valdymas -> Skaidrių įkėlimas -> Skaidrių įkėlimas iš zip archyvo

**Aktoriai:** *Vartotojas*

**Aprašas:** Įkelia skaidres

**Prieš-sąlygos:** nėra

**Sužadinimo sąlyga:** įkėlimo iš zip archyvo mygtuko parinkimo spragtelėjimas

**Po sąlyga:** skaidrių sąrašas paruoštas redagavimui, įrašo pradėjimui

9. PANAUDOJIMO ATVEJIS: Skaidrių valdymas -> Skaidrės pervadinimas

**Aktoriai:** *Vartotojas*

**Aprašas:** Pervadina skaidrę

**Prieš-sąlygos:** sukeltos skaidrės

**Sužadinimo sąlyga:** dvigubas paspaudimas ant skaidrės pavadinimo skaidrių sąrašė

**Po sąlyga:** parvadinta skaidrė

10. PANAUDOJIMO ATVEJIS: Skaidrių valdymas -> Skaidrių padėties keitimas -> Skaidrės kėlimas aukštyn

**Aktoriai:** *Vartotojas*

**Aprašas:** Pakeičia skaidrės poziciją į aukščiau esančią

**Prieš-sąlygos:** sukeltos skaidrės

**Sužadinimo sąlyga:** skaidrės perkėlimas perstumiant pele skaidrių sąrašė

**Po sąlyga:** pakeistas skaidrių eiliškumas

11. PANAUDOJIMO ATVEJIS: Skaidrių valdymas -> Skaidrių padėties keitimas -> Skaidrės leidimas žemyn

**Aktoriai:** *Vartotojas*

**Aprašas:** Pakeičia skaidrės poziciją į žemiau esančią

**Prieš-sąlygos:** sukeltos skaidrės

**Sužadinimo sąlyga:** skaidrės perkėlimas perstumiant pele skaidrių sąrašą

**Po sąlyga:** pakeistas skaidrių eiliškumas

12. PANAUDOJIMO ATVEJIS: Įrašo eksportavimas į VIPs -> Skaidrių siuntimas

**Aktoriai:** *Vartotojas*

**Aprašas:** nusiųsti skaidres į VIPs sistemą

**Prieš-sąlygos:** padarytas įrašas

**Sužadinimo sąlyga:** eksportavimo funkcijos pasirinkimas

**Po sąlyga:** perkeltos skaidrės į VIPs

13. PANAUDOJIMO ATVEJIS: Įrašo eksportavimas į VIPs -> Įrašo laikų siuntimas

**Aktoriai:** *Vartotojas*

**Aprašas:** Nusiųnčia įrašo laikus į VIPs sistemą

**Prieš-sąlygos:** padarytas įrašas

**Sužadinimo sąlyga:** eksportavimo funkcijos pasirinkimas

**Po sąlyga:** perkelti įrašo laikai į VIPs

14. PANAUDOJIMO ATVEJIS: Įrašo importavimas iš VIPs -> Skaidrių parsisiuntimas

**Aktoriai:** *Vartotojas*

**Aprašas:** Parsisiunčia skaidres iš VIPs sistemos

**Prieš-sąlygos:** egzistuojantis įrašytas įrašas

**Sužadinimo sąlyga:** importavimo funkcijos pasirinkimas

**Po sąlyga:** įkeltos skaidrės iš VIPs

15. PANAUDOJIMO ATVEJIS: Įrašo importavimas iš VIPs -> Įrašo laikų parsisiuntimas

**Aktoriai:** *Vartotojas*

**Aprašas:** Parsisiunčia įrašo laikus iš VIPs sistemos

**Prieš-sąlygos:** egzistuojantis įrašytas įrašas

**Sužadinimo sąlyga:** eksportavimo funkcijos pasirinkimas

**Po sąlyga:** įkelti įrašo laikai iš VIPs

16. PANAUDOJIMO ATVEJIS: Įrašo skaidrės parodymo laiko keitimas

**Aktoriai:** *Vartotojas*



**Aprašas:** Pakeičia skaidrės parodymo laiką

**Prieš-sąlygos:** įrašytas ar įkeltas įrašas

**Sužadinimo sąlyga:** skaidrės pasirinkimas

**Po sąlyga:** pakeisti įrašo laikai

## 2.2. Funkciniai reikalavimai ir reikalavimai duomenims

Šioje skiltyje apžvelgiami funkciniai reikalavimai. Taip pat pateikiama reikalavimų duomenims schema.

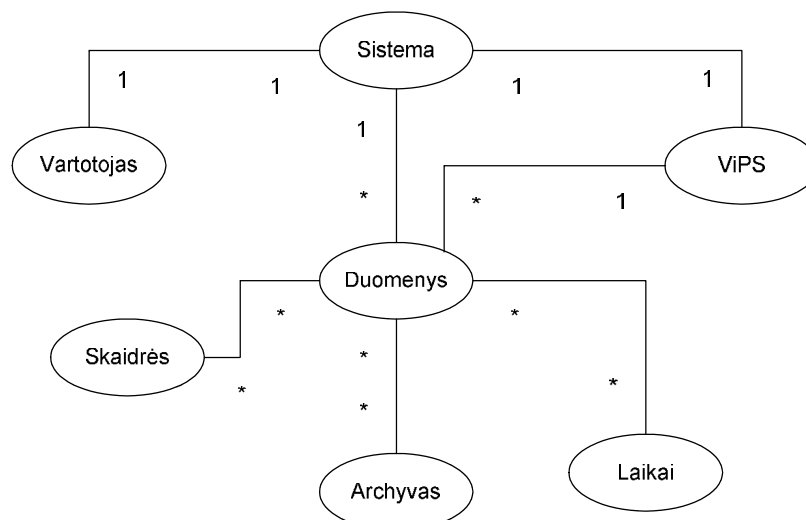
### 2.2.1. Funkciniai reikalavimai

Sistema turi išpildyti šiuos funkcinis reikalavimus:

- Skaidrių įkėlimas nurodžius katalogą
- Skaidrių įkėlimas nurodžius archyvą
- Skaidrių eksportas į ViPS
- Skaidrių sąrašo valdymas
- Skaidrės keitimas
- Skaidrės parodymo laiko keitimas
- Įrašo startavimas
- Įrašo stabdymas
- Įrašo importavimas iš ViPS

### 2.2.2. Reikalavimai duomenims

Žemiau pateiktame paveikslėlyje matome sistemos ir jos saryšio su duomenimis, vartotojais, bendradarbiaujančia sistema santikį.



7 paveikslėlis. Duomenų schema

## **2.3. Nefunkciniai reikalavimai**

Nefunkciniai reikalavimai susideda iš reikalavimų sistemos išvaizdai, panaudojamumui, vykdymo charakteristikoms, veikimo sąlygoms, priežiūrai ir saugumui.

### **2.3.1. Reikalavimai sistemos išvaizdai**

Sistemos išvaizda turi būti konkreti ir aiški. Tai reikalinga norint be vargo ir efektyviai naudotis programa. Tai bus realizuojama kuriant konkrečius mygtukų pavadinimus, bei parenkant tinkamą išdėstymą ir dydžius.

### **2.3.2. Reikalavimai panaudojamumui**

Sistemos funkcionalumas intuityviai suvokiamas ir valdomas. Reikalinga norint be vargo ir efektyviai naudotis programa.

### **2.3.3. Reikalavimai vykdymo charakteristikoms**

Veiksmai, įvykiai sistemoje turi trukti trumpai. Vartotojo laikas neturi būti gaišinamas, kadangi tai yra realaus laiko sistema.

### **2.3.4. Reikalavimai veikimo sąlygoms**

Sistema turi veikti ir senesniuose kompiuteriuose, kadangi ją gali prireikti naudoti įvairiausiose vietose, taip pat ir ten kur nėra patys naujausi kompiuteriai.

### **2.3.5. Reikalavimai sistemos priežiūrai**

Sistemos klaidas privaloma pakankamai greitai ištaisyti, nes tai apsunkina darbą su sistema, o būna ir atvejų kuomet visai nėra galimybės ją naudotis.

Įrankis turi veikti labiausiai paplitusiose pagrindinėse operacinėse sistemose, kadangi sistemos panaudojimą gali tekti įvairiose vietose su skirtingais kompiuteriais ir galbūt skirtingomis operacinėmis sistemomis.

### **2.3.6. Reikalavimai saugumui**

Sistema turi atlikti duomenų sinčiamų į ViPS kontrolę, taip pat neturi mažinti ViPS saugumo.

Sistemos darbo neturi trigdyti kitos sistemos ir pati programa turi veikti sklandžiai.

## **2.4. Architektūros specifikacija**

Šioje sekcijoje detaliai aprašoma produkto struktūra, kuri apžvelgiama iš kelių aspektų: dinaminio ir statinio, kelių tipu grafinėmis diagramomis. Tai pakankamai visapusiškas architektūros aprašas, leidžiantis gan išsamiai susipažinti su sukurta sistema.

### **2.4.1. Apžvalga**

Pateikta specifikacija sudaryta iš:

- Architektūros pateikimo – architektūros pateikimo būdas. Nurodoma, kokie yra reikalingi vaizdai ir kiekvienam vaizdui nurodoma, kokie modeliavimo elementai jį sudaro.
- Architektūros tikslų - programinės įrangos tikslų ir reikalavimų, turinčių esminį poveikį architektūrai aprašymas: COTS (commercial off-the-shelf) produktų panaudojimas, portabilumas, paskirstymas (distribution), pakartotinis panaudojimas (reuse) ir t.t.
- Sistemos statinio vaizdo - pateikiamas sistemos išskaidymas į paketus ir jų detalesnis aprašymas
- Sistemos dinaminio vaizdo - pateikiamos sąveikos (interaction), būsenų (state) ir veiklos (activity) diagramos.
- Išdėstymo vaizdo - aprašoma techninė įranga, kurioje sistema bus išdėstyta ir veiks, konfigūracija bei sistemos komponentai, išdėstyti atskiruose techninės įrangos mazguose.
- Duomenų vaizdo - pateikiamas duomenų bazės modelis
- Kokybės – architektūros įtakos aprašymas sistemos išplečiamumui, pernešamumui, patikimumui ir kitiems kokybės faktoriams, išskyrus funkcionalumą.

### **2.4.2. Architektūros pateikimas**

Architektūros specifikacija yra aprašyta naudojant UML standartizuotą modeliavimo ir specifikavimo kalbą, kuri naudoja grafinę notaciją abstrakčiam sistemos modeliui sukurti. Tai efektyvus ir konkretus būdas aprašyti sistemai, kadangi naudoja grafinius elementus modeliams kurti, kas aiškiau, bei konkrečiau apibrėžia sistemą ir nesudaro galimybės dviprasmiškam sistemos vaizdo supratimui. Architektūros vaizdavimui bus naudojamas Microsoft Office programų paketo „MS Visio Professional 2003“ įrankis, kuris leido sparčiau ir efektyviau sudaryti, bei modifikuoti UML diagramas. Sistemos architektūra bus pavaizduota iš kelių aspektų:

- Funkcinio - panaudojimo atviejų diagramos
- Objektinio – klasių diagramos
- Dinaminio – sekų, veikos, būsenų diagramos

### **2.4.3. Architektūros tikslai**

Tikslai architektūrai sudaryti iš reikalavimų: panaudojamumo, portabilumo, paskirstymo, pakartotinio panaudojimo.

#### **2.4.3.1. Tikslai ir Reikalavimai**

##### **Produkto panaudojimas**

- įrašinėjimu nesant interneto ryšiui
- įrašų ViPS sistemoje importavimas
- įrašų eksportavimas
- įrašų redagavimas.
- konkreti ir aiški įrankio vartotojo sąsaja
- sistemos funkcionalumas intuityviai suvokiamas ir valdomas
- veiksmi, įvykiai sistemoje neturi trukti ilgai
- sistema turi veikti ir senesniuose kompiuteriuose
- sistema turi atlikti duomenų sinčiamų į ViPS kontorlę
- sistemos darbo neturi trigdyti kitos sistemos

##### **Portabilumas**

- Sistemos klaidos turi būti pakankamai greitai ištaisomos
- Sistema turi veikti labiausiai paplitusiose pagrindinėse operacinėse sistemose

##### **Paskirstymas**

- Sistemą naudos tik naudojimo teisę turintys asmenys
- Pagal tam tikrą susitarimą su klientu leidimą sistemos naudojimui jis galės gauti

##### **Pakartotinis panaudojimas**

- Sistemos architektūra kuriama taip kad ją visą ar dalį jos būtų galima pakartotinai naudoti

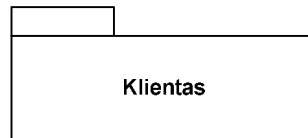
### **2.4.4. Sistemos statinis vaizdas**

Sistemos statinis vaizdas apima svarbias aplikacijos koncepcijas žvelgiant įskaitant realaus pasaulio koncepcijas, abstrakčias koncepcijas, realizavimo koncepcijas, kompiuterių koncepcijas – trupai tai yra visos koncepcijos randamos sistemose. Tai apima kuriamos

sistemos suskirstymą į paketus, detaliau paketuose arašomos klasės, interfeisai, jų ryšiai ir jų tipai, bei priklausomybės.

#### 2.4.4.1. Paketų detalizavimas

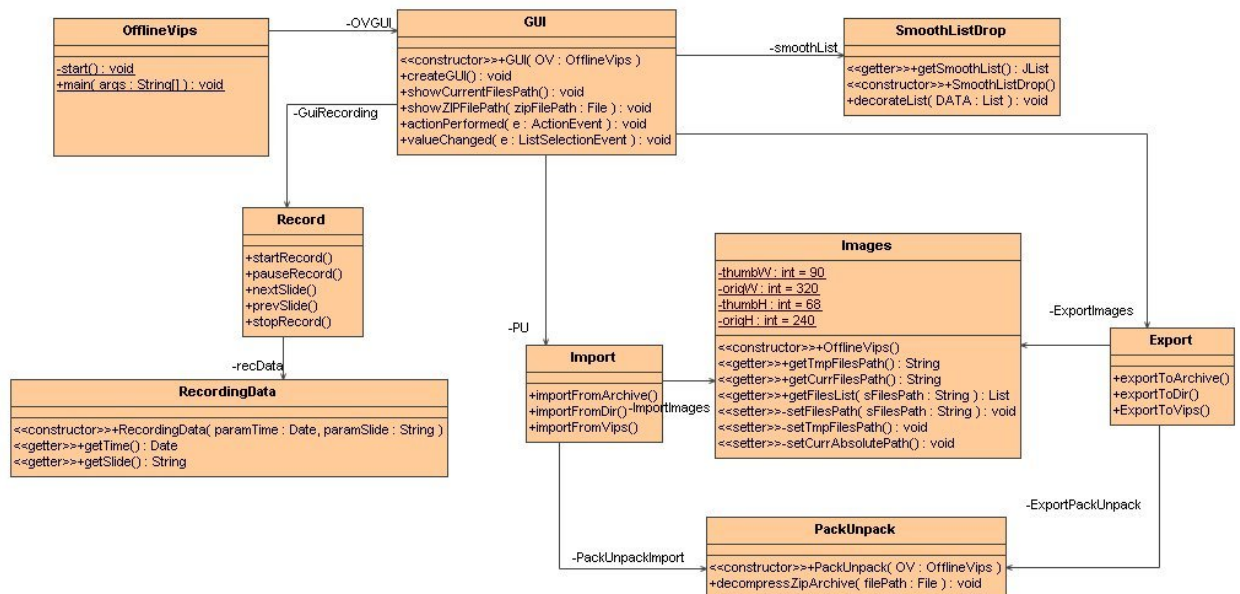
Mūsų sistema veikia kliento-serverio principu. Kadangi serverinė dalis realizuota, o taip pat apsibrėžtas tikslas sukurti klientą, tai sistema susidės iš vieno paketo – kliento paketo.



8 paveikslėlis. Sistemos paketas

#### 2.4.4.2. Klasių diagrama

Žemiau pateikiama sistemos klasių diagrama, braižyta panaudojant MagicDraw įrankį.(9 pav.)



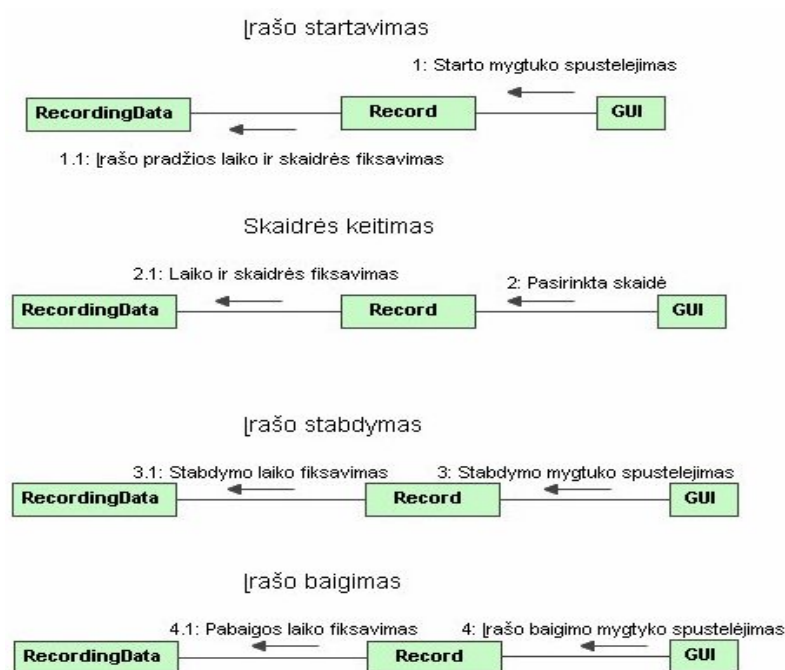
9 paveikslėlis. Klasių diagrama

#### 2.4.5. Sistemos dinaminis vaizdas

Sistemos dinaminis vaizdas pateikiamas bendradarbiavimo diagramomis.

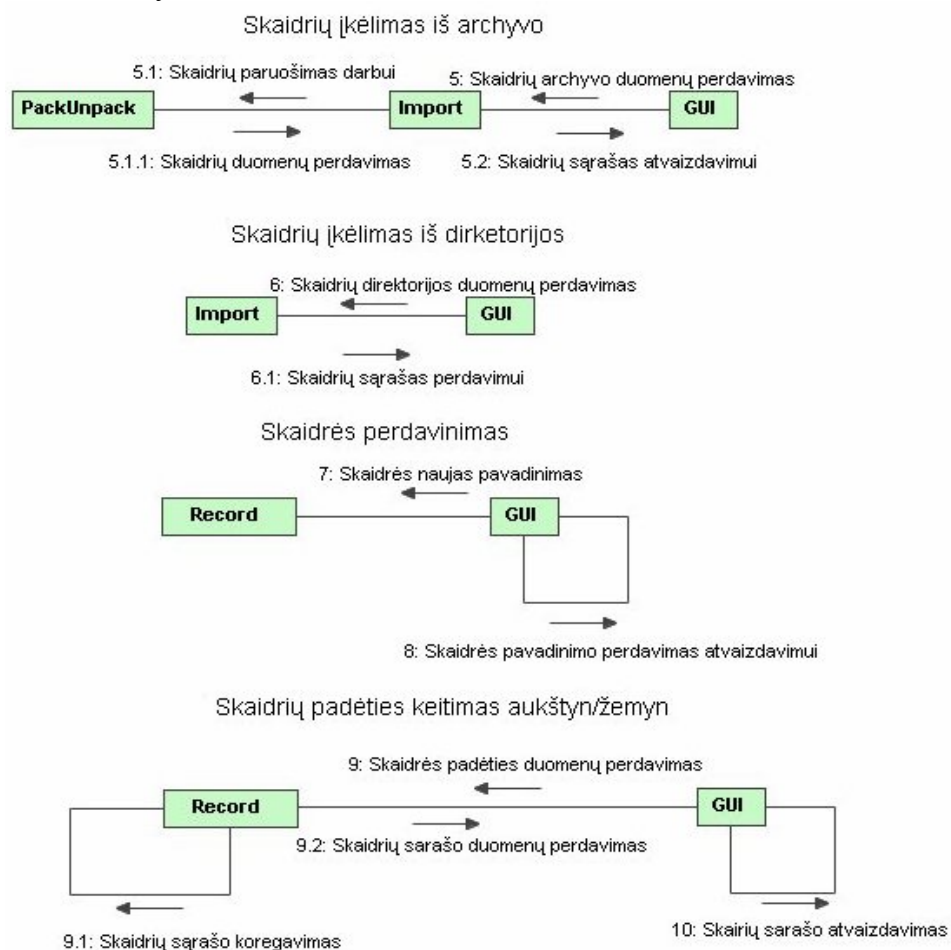
##### 2.4.5.1. Bendradarbiavimo diagramos

- Įrašo kontroliavimas



10 paveikslėlis. Įrašo kontroliavimo bendradarbiavimo diagramos

- Skaidrių valdymas



11 paveikslėlis. Skaidrių valdymo bendradarbiavimo diagramos

- Įrašo eksportavimas į VIPs ir įrašo importavimas iš VIPs

### Įrašo eksportavimas į VIPs



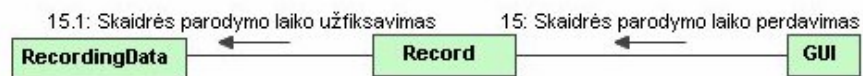
### Įrašo importavimas iš VIPs



**12 paveikslėlis.** Įrašo eksportavimo/importavimo bendradarbiavimo diagramos

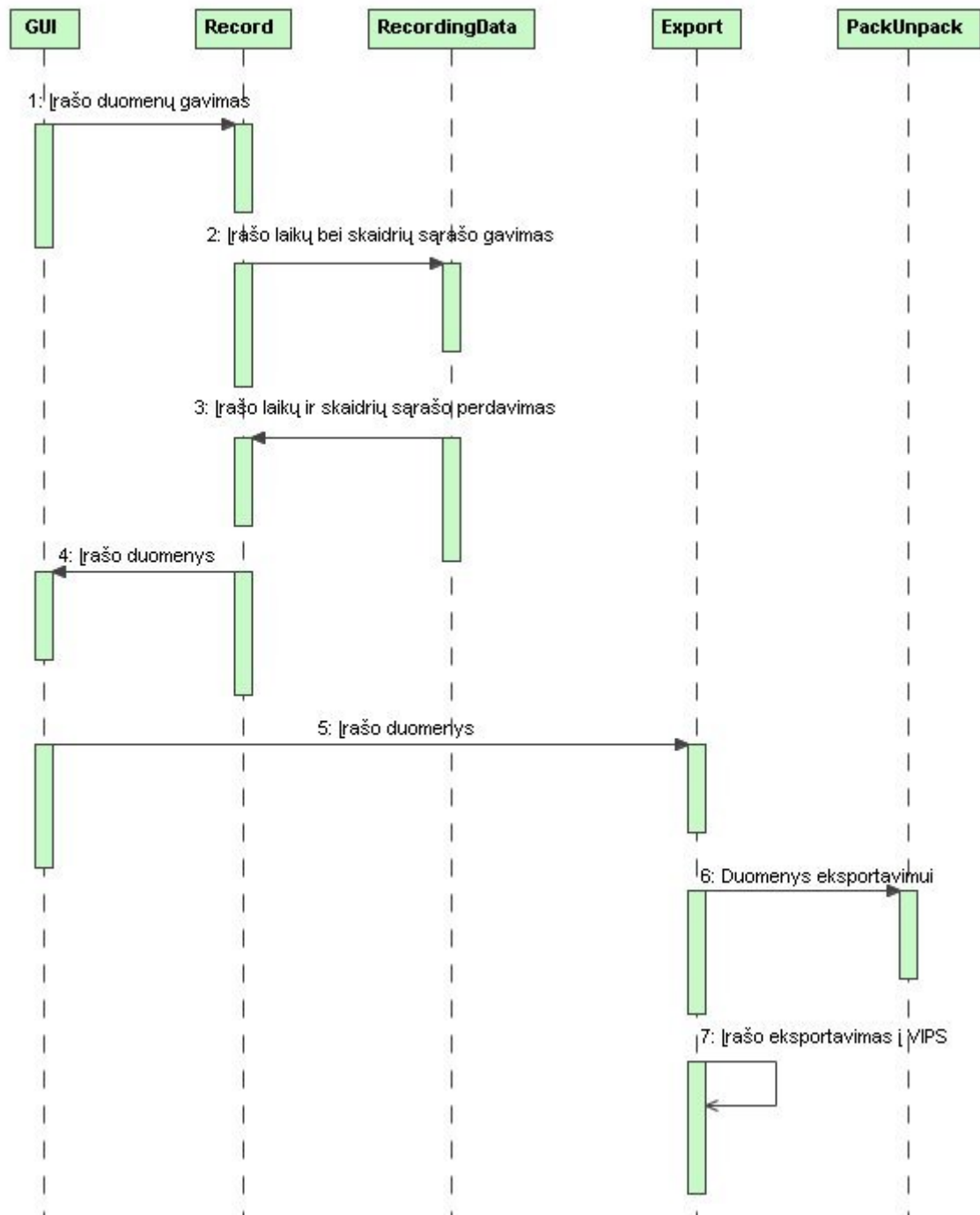
- Įrašo skaidrės parodymo laiko keitimas

### Įrašo skaidrės parodymo laiko keitimas



**13 paveikslėlis.** Įrašo skaidrės laiko keitimo bendradarbiavimo diagrama

- Sekų diagrama



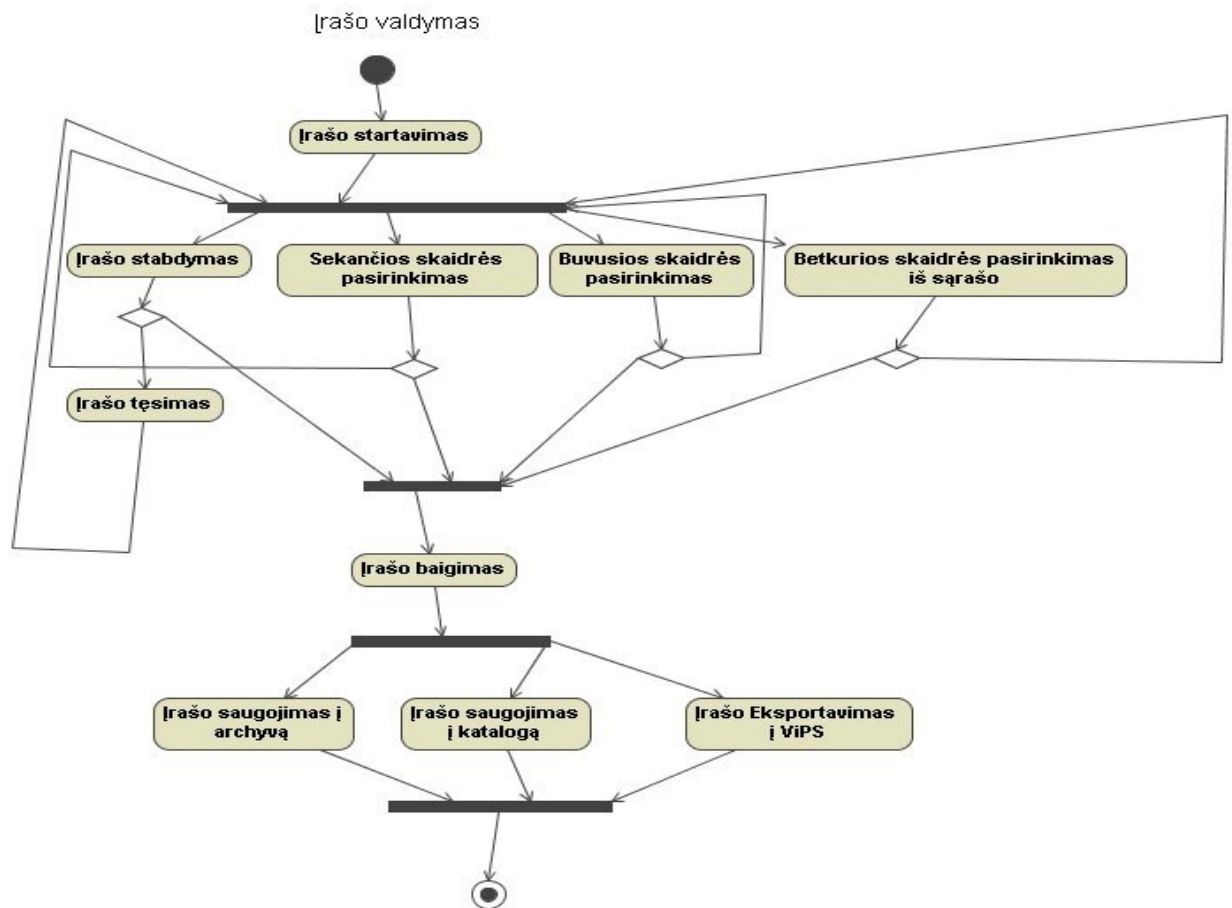
14 paveikslėlis. Sekų diagrama

### 2.4.5.2. Veiklos diagramos

Žemiau pateikiama įrašo valdymo, skaidrių valdymo, importavimo ir eksportavimo iš ViPS.

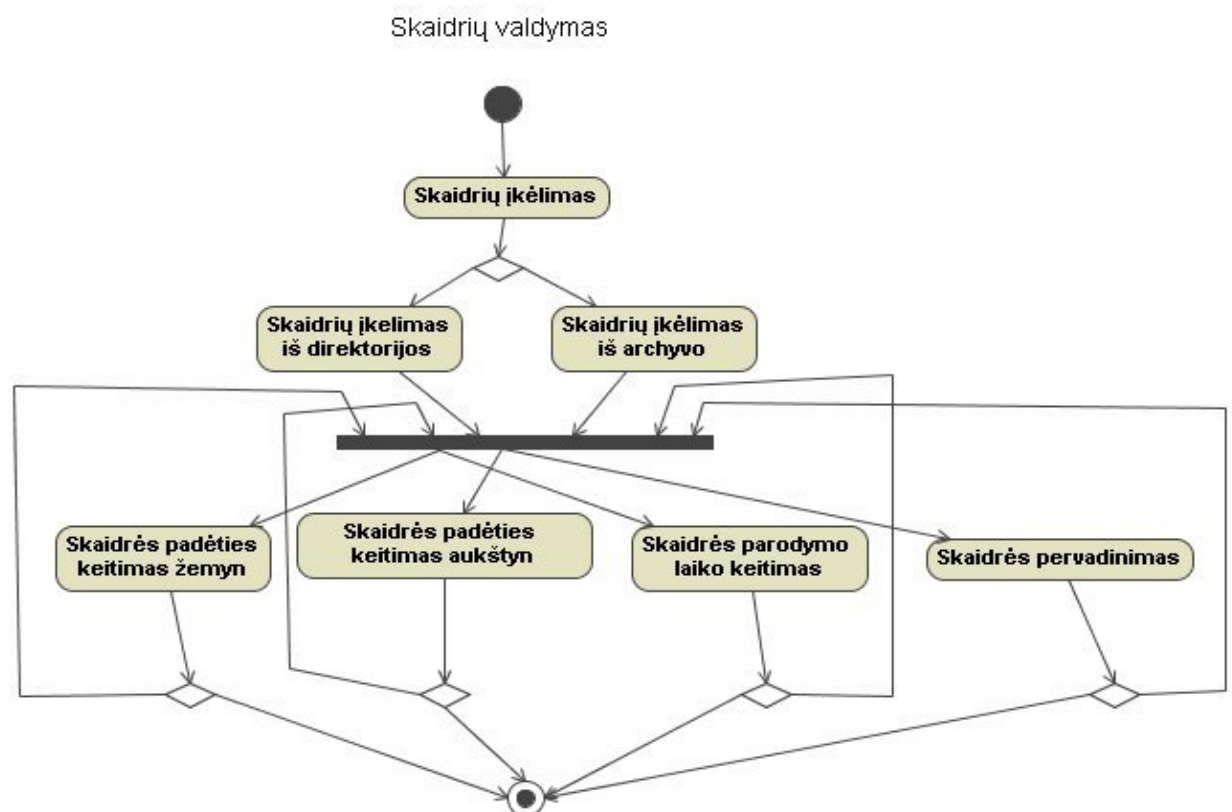
- Įrašo valdymas.(15 pav.)





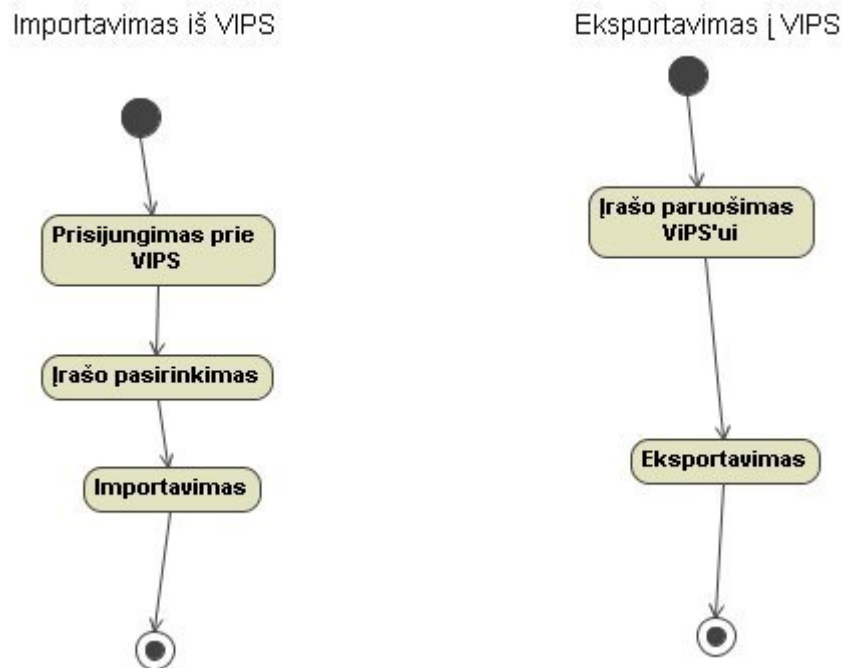
15 paveikslėlis. Įrašų valdymo veiklos diagrama

- Skaidrių valdymas(16 pav.)



16 paveikslėlis. Skaidrių valdymo veiklos diagrama

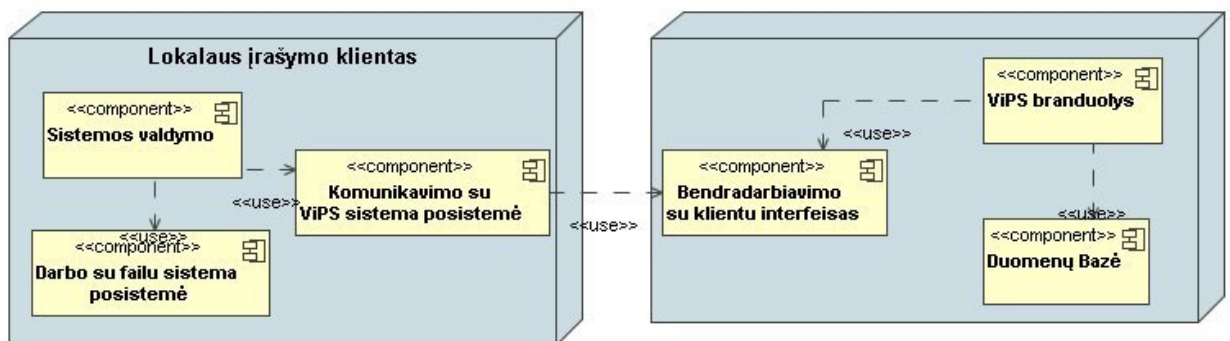
- Įrašo importavimas iš ir importavimas į ViPS(17 pav.)



17 paveikslėlis. Importavimo/Eksportavimo veiklos diagramos

#### 2.4.6. Išdėstymo vaizdas

Kuriama sistema yra suprojektuota veikti pagal kliento-serverio architektūrą. Jos komponentų išdėstymo vaizdas pavaizduotas paveiksle.(18 pav.)



18paveikslėlis. Išdėstymo vaizdas

#### 2.4.7. Duomenų vaizdas

Kadangi sistema pati tiesiogiai nenaudoja jokios duomenų bazės, o tik lokaliai atlieka įrašymą ir duomenis eksportuoja į ViPS sistemą kuri jau naudoja savo duomenų bazę, taigi duomenų bazės modelis nebus aptariamas.

#### 2.4.8. Kokybė

- Pernešamumas. Sistema gali būti naudojama daugumoje platformų, kadangi

reikalingas tik JAVA palaikymas.

- Išplečiamumas bei tobulinimas. Sistemos architektūra sukurta taip, kad nesukelia didelių problemų praplečiant ją naujomis galimybėmis.
- Patikimumas. Sistemos architektūra kurta taip, kad nereikėtų kartoti to pačio funkcionalumo keliose sistemos vietose, nes už tam tikra funkcionalumą yra atsakinga tik viena klasė.

## **2.5. Detalios architektūros specifikacija**

Ši specifikacija žvelgia giliau į sistemą ir labiau detalizuoja ją.

### **2.5.1. Kliento komponentas**

Šis paketas visas apima klases, jų operacijas ir kintamuosius. Tai vienintelis sistemos komponentas, kadangi nebuvo poreikio papildomiems komponentams kurti, jei bus poreikis, ateityje plečiant sistemą komponentų gali daugėti.

#### **2.5.1.1. Atsakomybės**

Atsakingas už visą sistemos veiklą, įskaitant:

- Įrašo valdymą
- Skaidrių įkėlimą
- Skaidrių valdymą
- Skaidrių importavimą
- Skaidrių eksportavimą
- Komunikaciją su pagrindine ViPS sistema
- Darbą su failų sistema

#### **2.5.1.2. Apribojimai**

Paketas turi formuoti įrašo duomenis taip kad jie tiktų importavimui į ViPS sistemą.

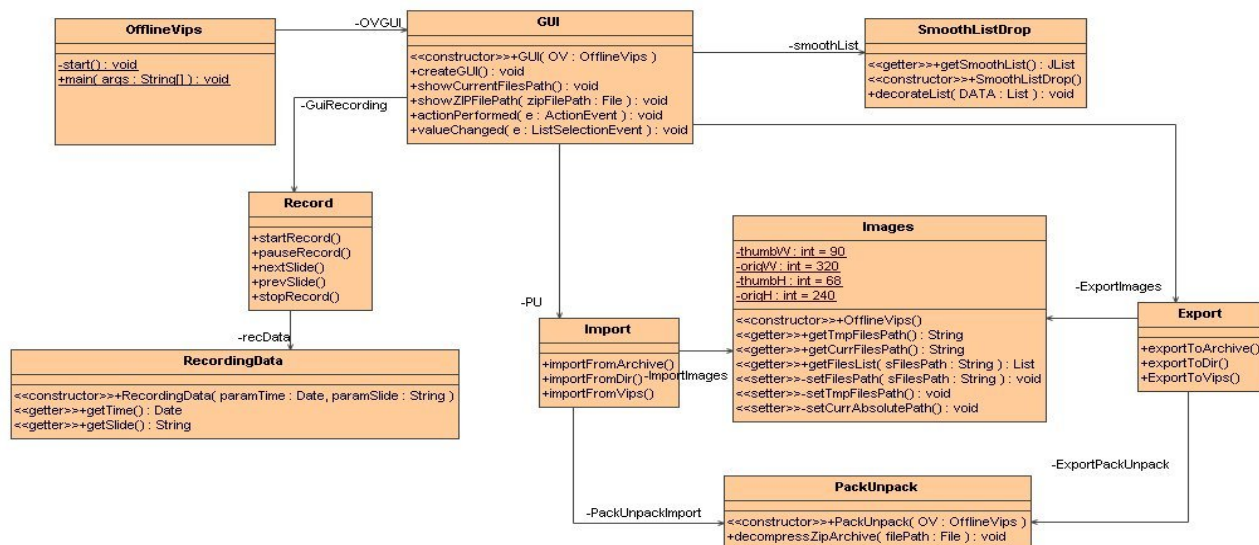
Paketas turi užtikrinti vartotojo perduodamų duomenų korektiškumą (neperduoti nepageidaujamų reikšmių, neleisti nepageidaujamų veiksmų sekų, perduoti vartotojo nustatytas reikšmes).

Pateikti vartotojui vienprasmiskai suprantamų veiksmų įvairovę.

Užtikrinti minimalų vartotojo klaidų galimybę (vartotojui suteikti duomenis rinktis iš reikšmių sąrašo, o ne laisvai įvedamų laukų, pateikti vartotojui ne tik pranešimus-įspėjimus apie neteisingus veiksmus, bet ir suteikti vartotojui pagalbą, kaip išvengti pakartotinio klaidos pranešimo).

### 2.5.1.3. Struktūra

Paketo struktūra pateikiama klasių diagramoje(19 pav.)



19 paveikslėlis. Paketo klasių diagrama

<b>Pavadinimas</b>	<b>OfflineVips</b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Inicializuoja programos pagrindines klases.
<b>Sąsaja</b>	Naudoja GUI klasę.
<b>Resursai</b>	Kompiuterio atmintis
<b>Sąveikavimas</b>	<i>main()</i> – pagrindinis sistemos medotas automatiškai paleidžiamas
<b>ir metodų</b>	startuojant programą
<b>aprašymai</b>	<i>start()</i> – startuoja sistemą, į šį medodą kreipiasi <i>main()</i> metodas.

<b>Pavadinimas</b>	<b>GUI</b>
<b>Klasifikacija</b>	Vartotojo sąsajos klasė
<b>Aprašymas</b>	Atvaizduoja grafinės vartotojo sąsajos komponentus, taip pat reaguoja į vartotojo veiksmus naudojantis sistema
<b>Sąsaja</b>	Naudoja <b>SmoothListDrop</b> , <b>Record</b> , <b>Import</b> , <b>Export</b> klases.
<b>Resursai</b>	Swing komponentų klasės, kompiuterio atmintis
<b>Sąveikavimas</b>	<i>createGUI()</i> – sukuria ir atvaizduoja vartotojo sąsajos komponentus
<b>ir metodų</b>	<i>showCurrentFilePath()</i> – parodo vartotojui tuo metu naudojamų skaidrių
<b>aprašymai</b>	buvimo vietą failų sistemoje <i>showZIPFilesPath()</i> – parodo vartotojui skaidrių archyvo buvimo vietą failų sistemoje <i>actionPerformed()</i> - realizuoja vartotojų pasirinktus veiksmus jiems

pateiktoje vartotojo sąsajoje

*valueChanged()* – vartotojui pasirinkus skaidrę iš sąrašo ją atvaizduoja

<b>Pavadinimas</b>	<b><i>SmoothListDrop</i></b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Sukuria animuotą ir akiai malonesnį skaidrių sąrašą, o taip pat suteikia galimybę pele keisti jų išdėstymą, bei pasirinkti skaidrę atvaizdavimui
<b>Sąsaja</b>	Su <b><i>GUI</i></b> klase.
<b>Resursai</b>	Kompiterio atmintis
<b>Sąveikavimas</b>	<i>getSmoothList()</i> – gražina skaidrių sąrašą
<b>ir metodų</b>	<i>decorateList()</i> – dekoruoja, pagražina sarašo atvaizdavimą
<b>aprašymai</b>	

<b>Pavadinimas</b>	<b><i>Record</i></b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Atsakinga už įrašą ir jo informaciją: startavimą, stabdymą, baiugimą. skaidrių eiliškumą, jų parodymo laikus, taip pat šių duomenų saugojimą
<b>Sąsaja</b>	<b><i>GUI, RecordingData</i></b> klasės
<b>Resursai</b>	Kompiuterio atmintis
<b>Sąveikavimas</b>	<i>startRecord()</i> – startuoja įrašą
<b>ir metodų</b>	<i>pauseRecord()</i> – laikinai sustabdo įrašą
<b>aprašymai</b>	<i>nextSlide()</i> – pateikia sekančią skaidrę <i>prevSlide()</i> – pateikia prieš tai buvusią skaidrę <i>cancelRecord()</i> – baigia įrašą

<b>Pavadinimas</b>	<b><i>RecordingData</i></b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Atsakinga už įrašo duomenų saugojimą.
<b>Sąsaja</b>	Su <b><i>Record</i></b> klase.
<b>Resursai</b>	Kompiuterio atmintis.
<b>Sąveikavimas</b>	<i>getTime()</i> – gražina skaidrės parodymo laiką
<b>ir metodų</b>	<i>getSlide()</i> – gražina skaidrę
<b>aprašymai</b>	<i>setData()</i> – užfiksuoja skaidrės rodymo laiką bei pačią skaidrę

<b>Pavadinimas</b>	<b><i>Import</i></b>
--------------------	----------------------

<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Atsakinga už įrašo importavimą iš ViPS, iš failų sistemos, iš archyvo
<b>Sąsaja</b>	<b>GUI, Images, PackUnpack</b> klasės
<b>Resursai</b>	Kompiuterio atmintis, failų sistema, nutolęs web serveris
<b>Sąveikavimas</b>	<i>importFromDir()</i> – importuoja iš direktorijos
<b>ir metodų</b>	<i>importFromArchive()</i> – importuoja iš archyvo failo
<b>aprašymai</b>	<i>importFromVips()</i> – importuoja iš ViPS

<b>Pavadinimas</b>	<b>Eksport</b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Atsakinga už įrašo eksportavimą į ViPS, į failų sistemą, archyvą
<b>Sąsaja</b>	<b>GUI, Images, PackUnpack</b> klasės
<b>Resursai</b>	Kompiuterio atmintis, failų sistema, nutolęs web serveris
<b>Sąveikavimas</b>	<i>exportFromDir()</i> – eksportuoja į direktoriją
<b>ir metodų</b>	<i>exportFromArchive()</i> – eksportuoja į archyvo failą
<b>aprašymai</b>	<i>exportFromVips()</i> – eksportuoja į ViPS

<b>Pavadinimas</b>	<b>PackUnpack</b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Atsakinga už archyvo sukūrimą bei išskleidimą
<b>Sąsaja</b>	<b>Import, Export</b> klasės
<b>Resursai</b>	Failų sistema, kompiuterio atmintis
<b>Sąveikavimas</b>	<i>decompressZipArchive()</i> – išarchyvuoja archyvo failą
<b>ir metodų</b>	<i>compressZipArchive()</i> – suarchyvuoja failus
<b>aprašymai</b>	

<b>Pavadinimas</b>	<b>Images</b>
<b>Klasifikacija</b>	Klasė
<b>Aprašymas</b>	Atsakinga už paveikslėlių formatą, jų dydžius.
<b>Sąsaja</b>	<b>Import, Export</b> klasės
<b>Resursai</b>	Kompiuterio atmintis.
<b>Sąveikavimas</b>	<i>getTmpFilesPath()</i> – gražina laikinosios direktorijos lokaciją
<b>ir metodų</b>	<i>getCurrFilesPath()</i> – gražina esamosios direktorijos buvimo vietą
<b>aprašymai</b>	<i>getFilesList()</i> – gražina failų sąrašą <i>setFilesPath()</i> – nustato failų direktoriją

*setTmpFilePath()* – nustato laikiną failų direktoriją

*setCurrAbsolutePath()* – nustato pagrindinę programos darbinę direktoriją

#### **2.5.1.4. Sąveikavimas**

Kliento paketas esant būtinybei komunikuoja tik su nutolusia ViPS sistema.

#### **2.5.1.5. Resursai**

Paketo naudojami resursai:

- Nutolusi ViPS sistema
- Failų sistema

#### **2.5.1.6. Skaičiavimai**

Paketas skirtas lokaliai įrašinėti video paskaitas, tiksliau jis atlieka skaidrių valdymo funkciją, vėliau viskas susinchronizuoja su video įrašu. Komponentas veikia kaip sistema ir jo veikimas priklauso nuo vartotojo pasirinktų veiksmų, tai gali būti skaidrių įkėlimas į sistemą iš:

- Direktorijos
- Archyvo
- Importuojant iš ViPS sistemos

Sudėtingi ar kelių žingsnių algoritmai nėra naudojami, pats darbas su sistema sukuria savitą veiksmų algoritmą. Galima būtų paminėti kad eksportuojant į ViPS iš pradžių sukuriamas tekstinis failas su skaidrių parodymo laikais, po to kartu su skaidrėmis padaromas vienas archyvo failas ir naudojant HTTP protokolą POST metodo būdu siunčiamas į nutolusią ViPS sistemą. Analogiškai veikia ir importavimo algoritmas, tik jo vygdymas vyksta priešinga eksportui eiga.

#### **2.5.1.7. Sąsaja**

Paketas turi sąsają su nutolusia ViPS sistema. Sąsaja naudojama duomenų importui, eksportui tarp nutolusio serverio. Importo/Eksporto metu perduodamas archyvo failas kuriame yra skaidrių paveikslėlių failai, bei tekstinis failas kuriame surašyti skaidrių parodymo laikai.

### **2.6. Testavimo Medžiaga**

Testavimo medžiaga sudaryta iš tikslų, objektų, testavimo plano, programinės įrangos, sąsajų, resursų, įrankių, aplinkos ir procedūrų aprašymo.

### **2.6.1. Testavimo tikslai ir objektai**

Testavimo planas yra skirtas:

Aprašyti sistemos komponentus, kurių kokybė turi būti patikrinta.

Aprašyti reikalavimus testavimui.

Aprašyti testavimo strategiją.

Nustatyti testavimui reikalingų resursų kiekį.

Sudaryti testuojamų komponentų sąrašą.

Aprašyti testavimui reikalingą aplinką.

Šiame dokumente aprašomas Lokalaus vaizdo paskaitų įrašymo ir įrašų redagavimo ViPS sistemai įrankio(toliau įrankio) testavimas. Dokumente yra pateikiamos rekomendacijos testavimui ir jis turi būti naudojamas kaip pagrindas testuojant sistemą. Taip pat atsakoma į klausimus:

Kas turi būti testuojama;

Kokie turi būti vykdomi testai;

Kaip turi būti vykdomi testai;

Kaip turi būti aprašomi testai;

Kaip turi būti planuojami darbai;

Kaip turi būti nustatoma testavimo rezultatų kokybė;

Kas yra atsakingas už darbus.

### **2.6.2. Testavimo apimtis**

Čia apžvelgiamas bendras programų sistemos testavimo planas. Atliekant kuriamos programinės įrangos testavimą, bus naudojama keletas testavimo metodų. Sistema susideda iš vartotojo sąsajos bei sąsajos su nutolusia ViPS sistema, bei jos duomenų bazės sistema. Pateikiami skirtingų rūšių testai programinės įrangos kokybei užtikrinti. Pasirinkti toliau išvardinti testavimo būdai:

- Vienetų testavimas;
- Priėmimo testavimas;
- Integracinis testavimas;



- Naudotojo sąsajos testavimas;
- Automatinis testavimas;
- Regresinis testavimas;
- Juodos dėžės testavimas;
- Sistemos pertestavimas po klaidų ištaisymo.

### 2.6.3. Testavimo planas

Testavimo plano tikslas yra pateikti procedūras ir strategijas kuriomis naudojantis būtų galim ištestuoti kuriamą programinę įrangą ir užtikrinti užsibrėžtus kokybės reikalavimus.

### 2.6.4. Testuojama programinė įranga

Duomenys iš failų sistemos įrankiui pateikiami juos pasirinkus iš failų sistemos, naudodami įrašo kontrolės posistemę, vartotojai pradeda įrašą, jį stabdo, keičia skaidres, įrašo redagavimo posistemę naudotojai redaguoja įrašo laikus, nusiunčia jį į ViPS, taip pat parsisiuntę ir pakoregavę jį atnaujina.

### 2.6.5. Sąsajos

**Prisijungimo sąsaja.** Suteikia galimybę vartotojui prisijungti prie ViPS įrašų parsisiuntimui, nusiuntimui, norint juos redaguoti ar patalpinti juos video paskaitų sistemoje. Naudotojas prisijungia įvedęs vartotojo vardą ir slaptažodį.

**Įrašo kontrolės sąsaja.** Vartotojas atlieka įrašo valdymą, įkelia skaidres, keičia jų eiliškumą, pavadinimus, startuoją įrašą, jo metu keičia rodomą skaidrę, stabdo, bei užbaigia įrašymą.

**Įrašo redagavimo sąsaja.** Naudotojas persisiunčia reikiamą įrašą, redaguoja skairių pasirodymo laikus, taip pat eiliškumą, pervadina skaidres, atnaujina ar nusiunčia naują įrašą.

### 2.6.6. Testavimo resursai

#### Klientas:

AMD Athlon™ 64 3200+ 2.01GHz, 1GB RAM

Microsoft Windows XP Professional, Service Pack 2.

Java SE 1.5 SDK

#### Serveris:

AMD Athlon™ 64 X2 Dual Core 4600+, 4GB RAM

Linux Fedora 8

Apache 2.0

PHP 5.1.6

MySQL 5.0.27

### 2.6.7. Testavimo įrankiai ir aplinka

Testavimui skirta bendra techninė ir programinė įranga pateikta skyrelyje 2.3.

Testavimui bus naudojami žemiau pateikti įrankiai:

- Vienetų testavimas
  - JUnit (<http://www.junit.org/>)
- Integravimo testavimas
  - Jamelon (<http://jameleon.sourceforge.net/index.html>)
- Regresinis testavimas
  - XTest (<http://xtest.netbeans.org/>)
- Stress testavimas
  - P-unit (<http://p-unit.sourceforge.net/>)
- Vartotojo sąsajos testavimas
  - Marathon (<http://marathonman.sourceforge.net>)

### 2.6.8. Testavimo procedūra

Testavimo procedūra sudaryta iš vienetų, integracijos, priėmimo, aukšto lygio testavimo, kiekvienas iš jų aprašytas žemiau.

#### 2.6.8.1. Vienetų testavimas

Šio proceso metu, testuojama „baltos dėžės“ principu, t.y. testiniai atvejai gaunami iš programos struktūros. Vienetų testavimas naudojamas atskiriems programinės įrangos komponentams testuoti. Komponentai testuojami paduodant įėjimo duomenis, stebint išėjimus ir juos lyginant su laukiamais rezultatais.

Šiame skyriuje aprašomi testavimų metodai, kurie bus atliekami kuriamai programinei įrangai testuoti, taip pat pateikiama testavimo procedūra bei aprašomi laukiami rezultatai.

**Prisijungimo sąsaja.** Sąsajos testavimų rezultatai(3 lentelė).

**3 lentelė.** Funkcijos public Boolean doLogin(String uname, String upass) testavimas

Įeinantys duomenys	Rezultatas kurio tikimasi	Testavimo
--------------------	---------------------------	-----------

		<b>rezultatas</b>
Prisijungimo vardas: "admin" Slaptažodis: ""	FALSE	Kaidos nėra
Prisijungimo vardas: "admin" Slaptažodis: "slaptazodis"	TRUE	Kaidos nėra
Prisijungimo vardas: "" Slaptažodis: "slaptazodis"	FALSE	Kaidos nėra
Prisijungimo vardas: "" Slaptažodis: ""	FALSE	Kaidos nėra

**Įrašo kontrolės sąsaja.** Sąsajos testavimų rezultatai(4,5,6 lentelės).

**4 lentelė.** Funkcijos public List addSlides(path) testavimas

<b>Įeinantys duomenys</b>	<b>Rezultatas kurio tikimasi</b>	<b>Testavimo rezultatas</b>
Tuščias katalogas	slidesList.size() == 0	Kaidos nėra
Katalogas su failais, kurie nėra paveikslėliai	slidesList.size() == 0	Kaidos nėra
Katalogas su paveikslėliais	slidesList.size() > 0	Kaidos nėra
Failas	NULL	Kaidos nėra

**5 lentelė.** Funkcijos public boolean changeSlidesPosition(int firstSlideIndex, int secondSlideIndex) testavimas

<b>Įeinantys duomenys</b>	<b>Rezultatas kurio tikimasi</b>	<b>Testavimo rezultatas</b>
-1, 0	FALSE	Kaidos nėra
-1, -10	FALSE	Kaidos nėra
1, 100 (kai skaidrių mažiau nei nurodoma pozicija)	FALSE	Kaidos nėra
1, 10 (kai skaidrių užtenka)	TRUE	Kaidos nėra

**6 lentelė.** Funkcijos public boolean changeSlideName(int index, String name) testavimas

<b>Įeinantys duomenys</b>	<b>Rezultatas kurio tikimasi</b>	<b>Testavimo rezultatas</b>
-1, "Pavadinimas"	FALSE	Kaidos nėra
0, "Pavadinimas"	TRUE	Kaidos nėra

100, "Pavadinimas" (kai skaidrių mažiau nei nurodoma pozicija)	FALSE	Kaidos nėra
10, "Pavadinimas" (kai skaidrių daugiau nei nurodoma pozicija)	TRUE	Kaidos nėra
2, ""	FALSE	Kaidos nėra

**Įrašo redagavimo sąsaja.** Sąsajos testavimų rezultatai(7, 8, 9 lentelės).

**7 lentelė.** Funkcijos public boolean getRecord(int recordID) testavimas

Įeinantys duomenys	Rezultatas kurio tikimasi	Testavimo rezultatas
Egzistuojantis recorded ir tinklas veikia	TRUE	Kaidos nėra
Egzistuojantis recorded ir tinklas neveikia	FALSE	Kaidos nėra
-1	FALSE	Kaidos nėra

**8 lentelė.** Funkcijos public boolean changeTime(int slideIndex, SimpleDateFormat time) testavimas

Įeinantys duomenys	Rezultatas kurio tikimasi	Testavimo rezultatas
-1, [korektiška data]	FALSE	Kaidos nėra
-1, [nekorektiška data]	FALSE	Kaidos nėra
100, [korektiška data] (kai skaidrių mažiau nei nurodoma pozicija)	FALSE	Kaidos nėra
10, [korektiška data] (kai skaidrių užtenka)	TRUE	Kaidos nėra

**9 lentelė.** Funkcijos public boolean sendRecord() testavimas

Įeinantys duomenys	Rezultatas kurio tikimasi	Testavimo rezultatas
Kai įrašas yra	TRUE	Kaidos nėra
Kai įrašo nėra	FALSE	Kaidos nėra
Kai įrašas su klaidom	FALSE	Kaidos nėra

### 2.6.8.2. Integravimo testavimas

Naudojant vienetų testavimą bus testuojami individualūs programinės įrangos komponentai. Vėliau jie apjungiami ir testuojami naudojant integracinį testavimą. Bus naudojamas Bottom-up integracinis testavimas. Testuojant įrankį, bus testuojami visi esami apjungti komponentai.

Testuojant sukuriama paprogramė, kurios leidžia sukurti tam tikrus atvejus, kuomet įrankio veikimas gali sutrikti arba funkcionuoti netinkamai.

### 2.6.8.3. Priėmimo testavimas

Naudojant šį metodą programinė įranga bus demonstruojama klientui, klientas ją išbandys norint nuspręsti ar ji atitinka kliento poreikius, specifikaciją. Testavimas atliekamas peržiūrint kiekvieną programinės įrangos panaudojimo atvejį. Aptikus neatitikimus vartotojo norams, patikrinsim, ar esanti programinė įranga atitinka specifikaciją, esant specifikacijos ir poreikių neatitikimui, bus registruojamas sistemos koregavimo reikalavimas, žymesni ir dideni pakeitimai bus atliekami sekančioje įrankio versijoje. Jei programinė įranga neatitinka specifikacijos bus pildoma klaidų ataskaita.

### 2.6.8.4. Aukšto lygio testavimas

**Regresinis testavimas.** Atlikus pakeitimus programinėje įrangoje bus pakartotinai vykdomi seni testai, taip pat sukuriama nauji testai patikrinti naujų programinės įrangos galimybių funkcionavimo teisingumą. Seni testai neatitinkantys naujų programinės įrangos galimybių nebenaudojami.

**Stress testavimas.** Įrankis, kuomet vartotojas dirba su įrašų redagavimu, prieš tai atsisisūsdamas ar pakoreguotus duomenis siūsdamas į ViPS, bus bandoma apkrauti tinklo srautą taip stebint kaip vyksta duomenų nusiuntimas ar parsisiuntimas.

Įrašomi vartotojo veiksmai bandant atlikti dokumentuose nurodytus programinės įrangos panaudojimo atvejų scenarijus. Vėliau naudojant P-unit programinę įrangą automatiškai vykdomi įrašyti. Stebimas klaidų kiekis.

**Vartotojo sąsajos testavimas.** Vartotojo sąsaja testuojama imituojuojant naudojo veiksmus, bandant tokį veiksmų eiliškumą kokio nebuvo tikėtasi kuriant įrankį, testavimui naujamas *Marathon* programa.

### **3. ĮRANKIO STANDARTIZAVIMO GALIMYBIŲ TYRIMAS**

#### **3.1. Egzistuojantys standartai ir jų savybių analizė**

Šio darbo tikslas sukurti įrankį skirtą lokaliai video prezentacijų įrašymui, bei eksportavimui į internetu pasiekiamą sistemą ViPS publikavimui, o taip pat įvertinti sinchroninio mokymosi standartizavimo galimybę, parenkant tinkamiausią, bei efektyviausią sistemai standartą. Taigi nuotolinis mokymasis taip pat standartizuojamas, turime pagrindines keturias organizacijas kurios tai atlieka:

- AICC
- IEEE
- IMS
- ADL

##### **3.1.1. AICC**

AICC standartas buvo naudojamas kuri laiką, tačiau vėliau tapo mažai kam reikalingas. Turi keistoką struktūrą, kuria įtakojo pradinis tikas: failais paremta komunikacija. Pradžioje tai buvo skirta kursų mokymuisi, kurie paleidžiami iš vykdomojo failo, tuomet programa rodo mokomąją medžiagą besimokančiajam, saugo rezultatus diske ir pabaigoje nusiunčia juos į serverį. Išplitus internetui ir išpopuliarėjus tinklapiams buvo papildomai pridėtas našyklėms tinkamas formatas informacijos perdavimui failuose. Buvo nuspresta pakuoti failų turinį internetine forma, atliekant *POST* užklausą.

Standartas turi apribojimų:

- Sukurta architektūra nepalaikoma senesnių nei Internet Explorer 5.5 versijų bei jaunesnių nei Netscape Navigator 6.02.
- Taip pat egzistuoja saugumo problemos, kurios įtakoja serverio-kliento komunikaciją

##### **3.1.2. LOM**

LOM standartas sukurtas IEEE organizacijos ir skirtas aprašyti mokymosi objektams, kurie yra bet kokia elektroninė arba neelektroninė medžiaga kuri skirta švietimui, bei mokymuisi.

Meta duomenys yra informacija apie informaciją ir yra struktūrizuoti tokiu būdu, kad palengvintų resursų valdymą ir paiešką Internetu. Meta duomenų standartai turi būti sukurti, kad palaikytų funkcinį suderinamumą (apsikeitimą informacija) tarp skirtingų technologijų ir įrenginių bei vartotojų užduotos resursų paieškos Internetu.

Meta duomenų standartai Internetui yra mėginimas užpildyti spragas tarp išsamaus profesionalaus katalogizavimo ir laisvo dokumentų kūrimo Internetu. Praktiškai, šie meta duomenų standartai leidžia dokumentų kūrėjams ir resursų valdytojams aprašyti resursus detalizuotose kategorijose palengvinant paieškos užklausų vykdymą.

Šiame standarte, mokymosi objektai apibrėžiami tiek skaitmeniniai, tiek neskaitmeniniai, kurie gali būti naudojami švietimui, mokymuisi ir mokymui. Mokymosi objektų meta duomenys aprašo svarbiausias naudojamas mokymosi objektų charakteristikas. Šios charakteristikos gali būti grupuojamos į pagrindines, gyvavimo ciklo, meta-meta duomenų, mokymosi, techninės, teisines, sąryšines, anotacines ir klasifikavimo kategorijas.

Koncepcinė duomenų schema specifikuoja konkrečioje kategorijoje leistiną lingvistinę įvairovę aprašant tiek mokymo objektus, tiek meta duomenis. Ši koncepcinė duomenų schema apibrėžia duomenų elementus, kurie sudaro mokymosi objektų meta duomenis.

LOM standarto pagrindinis tikslas yra palengvinti mokymosi objektų paiešką, įvertinimą, priėmimą ir naudojimą. Šis standartas taip pat palengvina dalijimąsi ir keitimąsi mokymosi objektais, kuriant katalogus, kur saugoma informacija apie kultūrinių ir kalbinių kontekstų įvairovę, kur mokymosi objektai ir jų meta duomenys buvo panaudoti.

### **3.1.3. IMS LD**

IMS Global learning consortium organizacija yra sukūrusi IMS LD specifikaciją, kuri specifikuoja metakalbą, leidžiančią modeliuoti mokymosi procesus. Atsiradimo šaknys siejasi su EML(Educational Modeling Language), sukurta Open University Olandijoje. Šis standartas buvo sukurtas todėl, kad mokymasis internetu tapo per daug statiškas ir pateikiama medžiaga skirta tik vienam vartotojui, nesuteikiant galimybės kooperotis, bendradarbiauti ar sulaukti pagalbos. Tai atspindi pedagogines idėjas, tačiau tai realizuoti sudėtinga. Šis standartas apima ne tik turinio informacijos ir paprastą mokymosi sekos organizavimą, bet ir vartotojus įtraukia į skirtingą užduočių atlikimą bei jų roles mokymosi metu.

IMS LD metakalba skirta aprašyti mokymosi modelius, kuri teigia kad yra pedagogiškai neautrali, turima galvoje kad neįpareigoja jokiems mokymosi būdams. Specifikacija gali būti palyginta su vaidmenimis teatre:

- Žmonės turi skirtingas roles
- Rolės apima tam tikrus tikslus, siekius, juos įgyvendinant per mokymosi ir/arba pagalbos metodus.

- Mokymasis, pagalba atliekama per tam tikrą aplinką, kurią sudaro mokymosi objektai bei servais.

IMS LD sudaryta iš trijų lygmenų (A, B ir C), kiekvienas jų prieš tai buvusį išpletoja ir apima:

- A lygmuo aprašo pagrindinius metakalbos elementus
- B lygmuo leidžia naudoti pagrindines elementų savybes ir salygines struktūras
- C lygmuo suteikia galimybę naudoti pranešimus, perspėjimus, suteikia dinamiškumo atliekant mokymosi užduotis, kadangi sekantis mokymosi etapas priklauso nuo prieš tai buvusio ar kelių buvusių žingsnių.

Tai yra metodas aprašyti mokymosi tikslams ir strategijoms. Kadangi kalba aprašoma XML formatu, taigi įrankis palaikantis IMS LD specifikaciją gali vygydyti jam pateiktą mokymosi medžiagą.

### **3.1.4. SCORM**

SCORM standartas sukurtas ADL (Advanced Distributed Learning) organizacijos iniciatyva. Jis apmima:

- Turinio ir mokymosi objekto archyvavimą-pakavimą
- Aprašo aplikacijų programavimo sąsają(API) ir duomenų modelį, skirtą naudoti komunikacijai tarp turinio objektų ir mokymosi valdymo sistemų.
- Nusako kaip mokymosi sekos tarp mokymosi veiksmų yra apobrėžtos ir interpretojamos.
- Atitikimo reikalavimų sąrašą, kurie patikrinami naudojant ADL SCORM atitikimo testų rinkinį.
- Taip pat turinio aprašo faile naudojama ir LOM specifikacija naudojamų duomenų metaduomenims aprašyti.

SCORM standartas apibrėžia ryšius tarp įvairių organizacijų sistemų, siekiant sukurti visiškai universalią mokymosi sistemą, kaip mokymo valdymo sistemą (LMS) arba mokymosi turinio valdymo sistemą (CMS). Tai yra pateikiama tipinė infrastruktūra, leidžianti bendrai naudoti mokymosi objektus išskirstytoje mokymosi aplinkoje. LMS saugomi kurso meta duomenys, o kursą sudarantys autonominiai mokymo objektai kuriami tiksliai apibrėžiant sąsają su LMS.

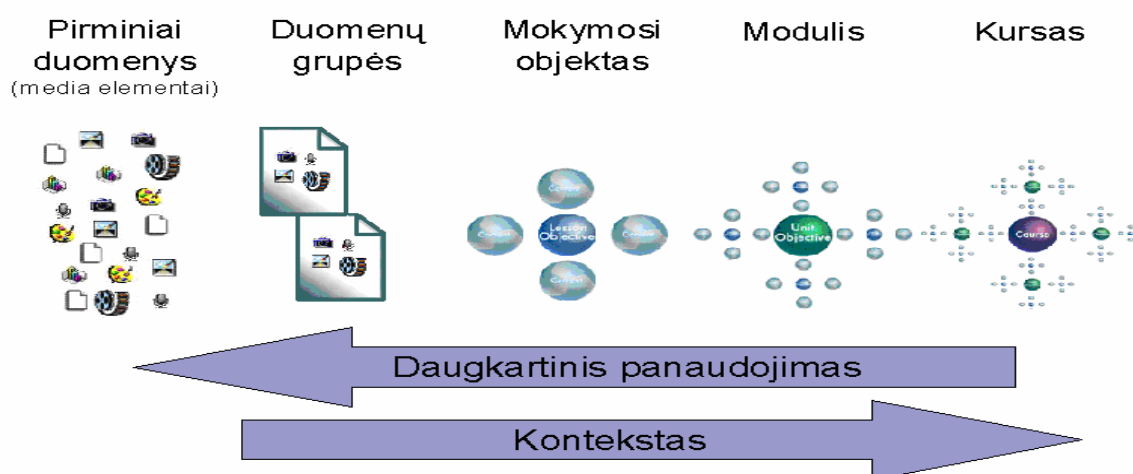
SCORM pateikia tarpusavyje susijusių standartų, specifikacijų bei instrukcijų rinkinį, užtikrinantį, kad mokymo valdymo sistemos ir mokymosi turinys atitiktų jiems keliamus aukštus reikalavimus. SCORM standartas apibrėžia šiuos reikalavimus kuriams mokymosi objektams:



- *Prieinamumas*: galimybė rasti ir prijungti kursų komponentus iš skirtingų vietų ir teikti juos kitiems.
- *Nepriklausomumas, mobilumas*: galimybė perkelti vienoje vietoje suprojektuotus ir sukurtus didaktinius komponentus kažkur kitur, ir naudoti juos skirtingose aplinkose su skirtingais įrankių rinkiniais.
- *Patvarumas*: sugebėjimas „atlaikyti“ technologinius pokyčius, išvengiant rekonfiguracijos, perprojektavimo ir perkodavimo.
- *Daugkartinis panaudojimas*: sugebėjimas apjungti suprojektuotus komponentus į skirtingas aplinkas ir kontekstus.

SCORM standarto pagrindinė idėja - yra tokia kursų konstrukcija, kuri leistų smulkinti žinias į mažus elementus, kurie gali būti pakartotinai panaudoti, ir tai vadinama bendrais turinio objektais – SCO. Jie atvaizduoja mokymosi objekto koncepciją su tam tikromis taisyklėmis, gaunamomis iš VMT. Šie objektai saugo informaciją, kuri gali būti saugoma naudojant tokius standartus, kaip HTML, XML ir kitus. Žinios, sukauptos kažkokiam e. kurse, gali būti panaudotos kituose kursuose, kadangi nėra jokių problemų išsirenkant ir panaudojant mokymosi objektus.

Pagal SCORM, mokymosi turinys sudaromas, pradedant nuo pačių smulkiausių objektų ir jungiant juos į blokus, tokius, kaip užduotys, skyriai, moduliai, kursai ir t.t. (**20 paveikslėlis.**). Kiekvienas turinio objektas neturi specifinio konteksto, taigi jį galima jungti su įvairiais kitais mokymosi turinio objektais. Šis principas reiškia, kad turinio objektas pats savaime nenulemia, kur ir kaip jis bus įjungtas į mokymosi bloką. Tai nusako *mokymo valdymo sistemos* taisyklės. Instruktoriai gali nurodyti nuoseklumo ir navigacijos tvarką, išlaikant galimybę panaudoti mokymosi resursus įvairiame kontekste.



**20 paveikslėlis.** Mokymosi turinio skaidymas ir grupavimas.

Paimta iš [http://gama.vtu.lt/KMS/KMS\\_teorija.doc](http://gama.vtu.lt/KMS/KMS_teorija.doc)

Specifikacijos rezultatai:

- Komunikacija tarp serverio ir kliento standartizuota:
  - Kursai gali naudoti betkokį grafinį dizainą
  - Galima kurti multimedijos elementus ir juos įtraukti į mokymosi medžiagą. Vienintelis reikalavimas yra tai, kad rezultatai būtų įtraukti į AICC/SCORM duomenų modelį.
- Dinaminis (serverio generuotas) turinys nėra leistinas:
  - Visi failai naudojami kursuose turi būti sukurti prieš sukuriant mokymosi medžiagą.
  - Tai leidžia mokymosi medžiagos transportavimą tarp serverių ir nepririša prie tam tikros platformos.

### **3.2. Tinkamumo ir kokybės kriterijų įvertinimas**

AICC siūlomas standartas kaip matyti turi trūkumų, kurie užkerta kelią pateikti mokymosi medžiagą vartotojui, todėl jis nėra tinkamas. LOM standartas skirtas metaduomenims aprašyti mokymosi proceso jis neorganizuoja, jį galima naudoti tik informacijai apie mokymosi objektus aprašyti.

Apibendrinant SCORM standartas skirtas apimti mokymosi objektams, tokiems kaip multimedia elementai, paveikslėliai, animacijos, HTML tinklapiai ir kiti turinio fragmentai, bei organizuoti mokymosi procesą.

IMS LD standartas buvo sukurtas tam, kad būtų galima realizuoti tas mokymosi proceso dalis, kurių neapima arba jos yra problematiškos SCORM standarte, deja IMS standarto problema ta kad jo realizacijos metodai per mažai specifikuotas, kas sumažina bet kurio standartizavimo aspekto vertę.

ViPS video paskaitų sistema informaciją besimokančiajam pateikia video medžiaga, bei skaidrėmis, paskaitos metu vedantysis užsiėmimą turi glaimybę:

- Keisti rodomas skaidres
- Užduoti klausančiajam klausimą
- Nusiąsti žinutę paskaitos dalyviams

Taigi, mums IMS LD siūlo roles, mokymosi scenarijus, kai tuo tarpu SCORM mums suteikia galimybę archyvuoti tam tikra tvarka turinį, mokymosi scenarijų kūrimą, bei nusako kaip ir koku būdu mokymosi objektai bus naudojami ir pateikiami vartotojui mokymosi sistemoje. Įvertinus abiejų standartų savybes matome jog tinkamesnis yra SCORM standartas.

### **3.3. Pasirinkto standarto realizacijos rekomendacijos įrankiui**

SCORM specifikacija sudaryta iš trijų pagrindinių dalių:

1. Turinio archyvavimo
2. Mokymosi aplinkos veikimo aprašymo
3. Medžiagos pateikimo eiliškumo organizavimo

Šiuo atveju standarto realizavimas įrankyje apims pirmąją ir paskutinąją dalis, kadangi pati mokymosi aplinka yra ViPS.

#### **3.3.1. Turinio archyvavimas**

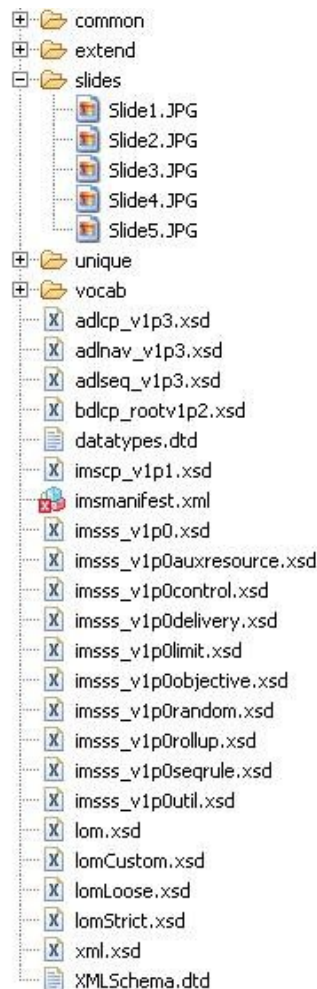
Mokymosi medžiagos turinys archyvuojamas į *zip* formato archyvą, o taip pat turinys yra palaikantis SCORM, kai:

- Gali būti perduotas per interneto naršyklę
- Nepriklausomas nuo serveryje vykdomų programavimo kalbų
- Nėra priklausomas nuo išorinių failų ar internetinių adresų
- Nepriklausomas nuo jokių parsisiunčiamų komponentų, kurie turėtų būti suinstaliuoti sistemos administratoriaus

Archyvo turinį sudaro:

- Mokymosi objektai
- Pagrindinio XML failo *imsmanifest.xml*. Šis failas aprašo visą mokymosi medžiagą ir instrukcijas SCORM vykdymo aplinkai.
- XSD ir DTD failai aprašantys *imsmanifest.xml* failą.

Turinio archyvo struktūros pavyzdys(21 pav.):



21 paveikslėlis. Turinio struktūros pavyzdys

## XML failo imsmanifest.xml struktūra:

```
<?xml version="1.0" standalone="no" ?>

<manifest identifier="MANIFEST_IDENTIFIER" version="1.0"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3"
xmlns:imsss="http://www.imsglobal.org/xsd/imsss" xmlns:lom="http://ltsc.ieee.org/xsd/LOM"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
    http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd
    http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd
    http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd
    http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd
    http://ltsc.ieee.org/xsd/LOM lom.xsd">

  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>2004 3rd Edition</schemaversion>
    <lom:lom>
      <lom:general>
        <lom:title>
          <lom:string language="en">Pavyzdys</lom:string>
        </lom:title>
        <lom:language>en</lom:language>
        <lom:description>
          <lom:string language="en">SCORM turinio archyvavimo
pavyzdys</lom:string>
        </lom:description>
        <lom:keyword>
          <lom:string language="en">SCORM, ViPS</lom:string>
        </lom:keyword>
      </lom:general>
    </lom:lom>
  </metadata>
</manifest>
```

```

        <lom:structure>
            <lom:source>LOMv1.0</lom:source>
            <lom:value>linear</lom:value>
        </lom:structure>
    </lom:general>
    <lom:lifeCycle>
        <lom:version>
            <lom:string language="en">1.0</lom:string>
        </lom:version>
        <lom:status>
            <lom:source>LOMv1.0</lom:source>
            <lom:value>final</lom:value>
        </lom:status>
    </lom:lifeCycle>
    <lom:technical>
        <lom:format>text/html</lom:format>
    </lom:technical>
    <lom:rights>
        <lom:cost>
            <lom:source>LOMv1.0</lom:source>
            <lom:value>yes</lom:value>
        </lom:cost>
        <lom:copyrightAndOtherRestrictions>
            <lom:source>LOMv1.0</lom:source>
            <lom:value>yes</lom:value>
        </lom:copyrightAndOtherRestrictions>
        <lom:description>
            <lom:string language="en">Edvinas Maciulis</lom:string>
        </lom:description>
    </lom:rights>
</lom:lom>
</metadata>

<organizations default="ORG-797533">
    <organization identifier="ORG-797533" structure="hierarchical"
adlseq:objectivesGlobalToSystem="true">
        <title>Pavadinimas</title>
        <item identifier="ACT-508158" identifierref="RES-817654">
            <title>Pav 1</title>
            <metadata>
                <lom:lom>
                    <lom:general>
                        <lom:structure>
                            <lom:source>LOMv1.0</lom:source>
                            <lom:value>linear</lom:value>
                        </lom:structure>
                    </lom:general>
                    <lom:metaMetadata></lom:metaMetadata>
                </lom:lom>
            </metadata>
            <adlcp:timeLimitAction>continue,no
message</adlcp:timeLimitAction>
            <adlcp:dataFromLMS>00:00:00</adlcp:dataFromLMS>
            <adlnav:presentation>
                <adlnav:navigationInterface>
                    <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
                </adlnav:navigationInterface>
            </adlnav:presentation>
        </item>
        <item identifier="ACT-932931" identifierref="RES-588158">
            <title>Pav 2</title>

            <adlcp:timeLimitAction>continue,no
message</adlcp:timeLimitAction>
            <adlcp:dataFromLMS>00:00:10</adlcp:dataFromLMS>
            <adlnav:presentation>
                <adlnav:navigationInterface>
                    <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
                </adlnav:navigationInterface>
            </adlnav:presentation>
        </item>
        <item identifier="ACT-592129" identifierref="RES-265199">
            <title>Pav 3</title>
            <adlcp:timeLimitAction>continue,no
message</adlcp:timeLimitAction>
            <adlcp:dataFromLMS>00:00:20</adlcp:dataFromLMS>
            <adlnav:presentation>
                <adlnav:navigationInterface>
                    <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>

```

```

        </adlnav:navigationInterface>
    </adlnav:presentation>
</item>
<item identifier="ACT-995485" identifierref="RES-53943">
    <title>Pav 4</title>
    <adlcp:timeLimitAction>continue,no
message</adlcp:timeLimitAction>
    <adlcp:dataFromLMS>00:00:30</adlcp:dataFromLMS>
    <adlnav:presentation>
        <adlnav:navigationInterface>
            <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
        </adlnav:navigationInterface>
    </adlnav:presentation>
</item>
<item identifier="ACT-59640" identifierref="RES-118729"
parameters="00:00:40">
    <title>Pav 5</title>
    <adlcp:timeLimitAction>continue,no
message</adlcp:timeLimitAction>
    <adlcp:dataFromLMS>00:00:40</adlcp:dataFromLMS>
    <adlnav:presentation>
        <adlnav:navigationInterface>
            <adlnav:hideLMSUI>exit</adlnav:hideLMSUI>
        </adlnav:navigationInterface>
    </adlnav:presentation>
</item>

    </organization>
</organizations>

<resources>
    <resource identifier="RES-817654" type="webcontent" adlcp:scormType="sco">
        <file href="slides/Slide1.JPG"></file>
    </resource>
    <resource identifier="RES-522826" type="webcontent" adlcp:scormType="sco">
        <file href="slides/Slide2.JPG"></file>
    </resource>
    <resource identifier="RES-265199" type="webcontent" adlcp:scormType="sco">
        <file href="slides/Slide3.JPG"></file>
    </resource>
    <resource identifier="RES-29766" type="webcontent" adlcp:scormType="sco">
        <file href="slides/Slide4.JPG"></file>
    </resource>
    <resource identifier="RES-118729" type="webcontent" adlcp:scormType="sco">
        <file href="slides/Slide5.JPG"></file>
    </resource>
    <resource identifier="RES-588158" type="webcontent" adlcp:scormType="sco">
        <file href="slides/chat.flv"></file>
    </resource>
    <resource identifier="RES-53943" type="webcontent" adlcp:scormType="sco">
        <file href="slides/yesno.html"></file>
    </resource>
</resources>
</manifest>

```

### 3.3.2. Mokymosi medžiagos pateikimo eiliškumo organizavimas

ViPS sistema turi linijinį mokymosi medžiagos pateikimo mechanizmą, taigi mokymosi medžiaga perduodama paeiliui viena paskui kitą. Pagal imsmanifest.xml faile paduodamus laikus vygdymo aplinka atvaizduos atitinkamai reikiamą mokymosi objektą besimokančiajam medžiagą.

## 4. IŠVADOS

1. Atlikta ViPS sistemos analizė, išanalizuota struktūra, funkcijos, kūrimo principai.
2. Išanalizuoti konkurenciniai produktai. Pastebėti konkurencinių sistemų privalumai ir trūkumai, tinkamumas.

3. Sukurtas įrankis, gebantis įrašinėti video prezentacijas nesant interneto ryšiui.
4. Realizuota sąsaja su ViPS leidžianti perduoti informaciją.
5. Buvo pasirinkti testavimo metodai ir ištestuotos atskiros programinės įrangos funkcijos, sudaryta testavimo PĮ.
6. Praktinių bandymų metu paaiškėjo, jog sukurta sistema tenkina visus iškeltus vartotojo reikalavimus
7. Atlikta egzistuojančių nuotolinio mokymosi standartų apžvalga.
8. Kiekvienas standartas buvo išanalizuotas ir įvertintas.
9. Parinktas tinkamiausias standartas įrankiui, tiek pagal veikimo principus, tiek pagal naudingumą jį palaikant.
10. Pateikti realizacijos įrankio rezultatų pavyzdžiai.

## 5. TERMINŲ SANTRUMPŲ ŽODYNAS

**JAVA** – multipratforminė programavimo kalba

**HTTP** – hypertext transfer protocol (hypertekstinio perdavimo protokolas)

**FTP** – file transfert protocol (failų persiuntimo protokolas)

**Flash** – Adobe Systems sukurta vektorinės grafikos animacijos technologija

**Shockwave** – programinė įranga skirta kurti ir platinti animacijos turinčios multimedijos elementų internete

**GIF** – graphics interchange format (vienas iš paveikslėlių formatų palaikantis 256 spalvas)

**ViPS** - Video paskaitų sistema

**PĮ** - Programinė įranga

**PA** – Panaudojimo atviejis

**UML** - Unifikuota modeliavimo kalba, naudojama objektiškai orientuotame projektavime (angl. Unified Modeling Language)

**XML** - Išplečiama žymių (arba teksto išdėstymo ir struktūrizavimo) kalba (angl. eXtensible Markup Language)

**CMS** – turinio valdymo aplinka

**LMS** – mokymosi valdymo aplinka

**OS** – operacinė sistema

**POST** – HTTP užklauso parametru perdavimo metodas

**ZIP** – archyvo failo tipas

**API** – apilikacijų programavimo sąsaja

**DTD** – dokumento tipo aprašymo formatas

## 6. LITERATŪRA

1. WebCT sistema, [Žiūrėta 2006-12-02]. Prieiga per internetą <http://www.webct.com>.
2. Moodle sistema, [Žiūrėta 2006-12-02]. Prieiga per internetą <http://moodle.org>.
3. ViPS sistema, [Žiūrėta 2006-12-02]. Prieiga per internetą <http://distance.ktu.lt/vips>
4. CDK įrankis, [Žiūrėta 2006-12-02]. Prieiga per internetą <http://distance.ktu.lt/cdk>
5. WebReg sistema, [Žiūrėta 2006-12-02]. Prieiga per internetą <http://distance.ktu.lt/kursai/webreg>
6. Mačiulis E., Jarušaitis A., WebReg registracijos ir apklausos sistemos panaudojimas nuotoliniame švietime. Informacinės technologijos 2006.
7. Cibulskis G., Balbieris G., Siegas M., Meškauskas D. Vaizdo paskaitų sistemos ViPS funkcinių galimybių plėtra. Informacinės technologijos 2005.
8. Cibulskis G., Siegas M. Video-lectures over Internet: development of tools and services // TELDA'03 Informacinėmis technologijomis grįstas mokymasis skaitmeniniame amžiuje : tarptautinė konferencija, 2003
9. Centra įrankis, [Žiūrėta 2006-12-07]. Prieiga per internetą <http://www.saba.com/products/centra>
10. Horizon Wimba sistema, [Žiūrėta 2006-12-09]. Prieiga per internetą <http://www.horizonwimba.com>
11. Interwise sistema, [Žiūrėta 2006-12-10]. Prieiga per internetą <http://www.interwise.com/>
12. Microsoft produkcija, [Žiūrėta 2006-12-08]. Prieiga per internetą <http://www.microsoft.com>
13. Hypertext Transfer Protocol -- HTTP/1.1, [Žiūrėta 2006-12-15]. Prieiga per internetą <ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt>
14. File Transfer Protocol, [Žiūrėta 2006-12-15]. Prieiga per internetą <ftp://ftp.rfc-editor.org/in-notes/rfc959.txt>
15. JAVA technologijos, [Žiūrėta 2006-12-17]. Prieiga per internetą <http://java.sun.com>
16. N.Bevan. Quality in use for all. User interfaces for all. Stephanidis, C (ed), Lawrence Erlbaum, 1999.
17. G.Gledec. Evaluating Web Site Quality. 2005.
18. J.Offutt. Web Software Applications QualityAttributes. Quality Engineering in Software Technology (CONQUEST 2002), pages 187-198, Nuremberg, Germany, September 2002.



19. Advanced Distributed Learning, SCORM 2004 3<sup>rd</sup> Edition Sharable Content Object Reference Model. 2006
20. IMS, IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata. 2006
21. AICC standartas, [Žiūrėta 2008-03-16]. Prieiga per internetą  
<http://www.readygo.com/aicc/>
22. IMS, IMS Learning Design Best Practice and Implementation Guide, 2003