

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

**Ričardas Liekis**

**ELEKTRONINIŲ KURSŲ MOKYMO OBJEKTŲ  
KOMPONAVIMO SISTEMA**

Magistro darbas

**Vadovas  
doc. dr. V.Šakys**

**KAUNAS, 2006**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

# **ELEKTRONINIŲ KURSŲ MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMA**

Magistro darbas

Vadovas

doc. dr. V.Šakys

Recenzentas

doc. A. Lenkevičius

Atliko

IFM 0/4 gr. stud.

R. Liekis

**KAUNAS, 2006**

## SUMMARY

The subject of this job is „The Packing System of Electronical learning objects“.

The main purpose of this job consists of research what is necessary to make the electronical learning course become totally independent from LMS; of the Project and the realization of the tool, which would allow demonstrating the quality of creating this type of course.

Universal systems, able to create every necessary elements for the electronical learning allow to produce learning courses and other related services in operative, objective, professional ways. Therefore this is the reason why the flexibility, giving the choice of necessary and useful system components, install the innovations quickly, is being lost. That is why massive LMS are going to be assembled of different components.

The analysis shows that the best choice is to use SCORM standard, which is the main standard of electronical learning.

The jobs being fulfilled:

- The analysis of electronical learning;
- The project of the packing system of electronical learning objects;
- The realization of the packing system of electronical learning objects;
- The experiment of:
  - The action of the produced system;
  - The investigation of the learning objects package.

The main characteristics of The Packing System of Electronical learning objects:

- The packing of functional learning objects;
- Effective usage of shareable learning objects;
- The Standard XML structure of learning objects.

# TURINYS

ĮVADAS.....	7
1. ELEKTRONINIŲ KURSŲ KŪRIMO SISTEMŲ ANALIZĖ .....	9
1.1. SISTEMAI KELIAMSI REIKALAVIMAI BEI NUMATOMAS FUNKCIONALUMAS.....	9
1.2. PROBLEMOS SPRENDIMŲ ANALIZĖ .....	10
1.3. ELEKTRONINIO MOKYMO PAGRINDAI.....	10
1.4. ELEKTRONINIS MOKYMAS LIETUVOJE IR UŽSIENYJE.....	11
1.5. ELEKTRONINIŲ MOKYMO SISTEMŲ KŪRIMO PRINCIPŲ IR PROBLEMŲ ANALIZĖ.....	12
1.6. GRĮŽTAMASIS RYŠYS .....	13
1.7. ELEKTRONINIO MOKYMO MEDŽIAGOS RUOŠIMAS .....	14
1.8. PEDAGOGO (MOKYTOJO) VEIKLA ELEKTRONINIAME MOKYME.....	15
1.9. ELEKTRONINIŲ KURSŲ KŪRIMO APLINKOS ANALIZĖ .....	16
1.9.1. VARTOTOJŲ TIKSLAI IR PROBLEMOS.....	17
1.9.2. ELEKTRONINIO MOKYMO SISTEMOS BRUOŽAI.....	17
1.9.3. ELEKTRONINIO MOKYMO KURSŲ KŪRIMO SISTEMOS.....	18
1.9.4. DABARTINIAI ELEKTRONINIO MOKYMO STANDARTAI.....	18
1.9.5. MOKYMO TURINIO VALDYMO SISTEMŲ ANALIZĖ.....	20
1.10. SCORM STANDARTAS .....	21
1.10.1. SCORM PAGRINDAI.....	21
1.11. ELEKTRONINIO MOKYMO SISTEMOS FUNKCINIS MODELIS .....	24
1.12. ELEKTRONINIŲ KURSŲ MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMOS SPECIFIKAVIMAS .....	24
1.13. KOMPIUTERIZUOJAMOS SISTEMOS FUNKCIJOS .....	26
1.14. FUNKCINIAI IR NEFUNKCINIAI REIKALAVIMAI,                    APRIBOJIMAI .....	27
1.14.1. REIKALAVIMAI STANDARTAMS .....	28
1.14.2. REIKALAVIMAI VARTOTOJO SĄSAJAI.....	28
1.14.3. REIKALAVIMAI DUOMENIMS .....	28
1.15. KŪRIMO ĮRANKIŲ PASIRINKIMAS. ....	29
1.16. DARBO TIKSLAS IR SIEKIAMI PRIVALUMAI .....	29
1.17. ANALIZĖS IŠVADOS .....	30
2. ELEKTRONINIŲ KURSŲ MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMOS PROJEKTAVIMAS .....	31
2.1. KOMPONAVIMO PROCESO DETALIZAVIMAS .....	31
2.2. KOMPONAVIMO SISTEMOS ELGSENOS APRAŠYMAS.....	33
2.3. SCORM STANDARTO REIKALAVIŲ IŠSKYRIMAS.....	35
2.3.1. MOKYMO OBJEKTŲ ORGANIZAVIMAS IR ATVAIZDAVIMAS.....	36
2.3.2. METADUOMENŲ APRAŠYMAS IR MANIFESTO FAILO GENERAVIMAS.....	38
2.3.3. MOKYMO OBJEKTO TURINIO PAKAVIMAS .....	38
2.4. SISTEMOS REIKALAVIMŲ APIBENDRINIMAS .....	38
2.5. DALYKINĖS SRITIES PRADINĖ PAKETINĖ ARCHITEKTŪRA .....	39
2.6. KOMPONAVIMO SISTEMOS PAKETINĖ ARCHITEKTŪRA .....	40
2.6.1. TURINIO PAKAVIMO POSISTEMĖ .....	41
2.6.2. METADUOMENŲ APRAŠYMO POSISTEMĖ .....	41
2.7. REALIZACIJOS MODELIS .....	43
2.8. ELEKTRONINIŲ KURSŲ KOMPONAVIMO SISTEMOS VARTOTOJO SĄSAJOS MODELIS .....	44
2.9. PROJEKTAVIMO IŠVADOS .....	46
3. EKSPERIMENTINIS TYRIMAS.....	47
3.1. EKSPERIMENTO APRAŠYMAS .....	47

3.2.	ELEKTRONINIO KURSO STRUKTŪRA IR PARUOŠIMAS MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMOS APDOROJIMUI.....	47
3.3.	SISTEMOS VEIKIMO EKSPERIMENTINIS TYRIMAS.....	48
3.4.	EKSPERIMENTINIO ELEKTRONINIO KURSO TYRIMAS.....	49
3.4.1.	ELEKTRONINIO KURSO ĮKĖLIMO TESTAVIMAS.....	49
3.4.2.	ELEKTRONINIO KURSO ATVAIZDAVIMO TEISINGUMO TYRIMAS.....	51
3.4.3.	SCORM APRAŠYMO TEISINGUMO ANALIZĖ .....	55
3.5.	EKSPERIMENTO APIBENDRINIMAS (IŠVADOS) .....	55
4.	VARTOTOJO DOKUMENTACIJA .....	56
4.1.	PROGRAMOS APRAŠAS .....	56
4.2.	VARTOTOJO SAŠAJA.....	56
4.2.1.	PAGRINDINIUIOSE ŽINGSNIUIOSE ESANČIŲ MYGTUKŲ APRAŠYMAS .....	56
4.2.2.	PROGRAMĄ SUDARANTYS PAGRINDINIAI ŽINGSNIAI.....	57
4.2.3.	METADUOMENŲ APRAŠYMO POSISTEMĖ .....	60
4.3.	PROGRAMOS VEIKIMAS .....	63
4.4.	PROGRAMOS DUODAMAS REZULTATAS .....	64
5.	IŠVADOS.....	65
6.	LITERATŪRA.....	66
7.	SANTRUMPŲ IR TERMINŲ ŽODYNAS .....	67
8.	PRIEDAI.....	68
8.1.	PROGRAMOS IŠEITIES TEKSTO FRAGMENTAI.....	68
8.2.	EKSPERIMENTINIO ELEKTRONINIO KURSO XML STRUKTŪRA .....	70
8.3.	NAUDOJAMŲ XML APRAŠŲ PAIŠKINIMAS .....	73
8.4.	PRIDEDAMO CD TURINYS .....	74
8.5.	E-KURSO SAŲVEIKŲ TARP KLIENTO IR SERVERIO SCHEMOS .....	75

## LENTELIŲ SĄRAŠAS

1.1 lentelė – Panaudojimo atvejų paaiškinimas .....	27
1.2 lentelė – Sistemos privalumai .....	30
2.1 lentelė - Uždavinių-procesų aprašymas.....	31
2.2 lentelė - Elektroninio kurso mokymo objektų komponavimo sistemos paketų paaiškinimas .....	40
3.1 lentelė – Elektroninių kursų mokymo objektų komponavimo sistemos funkcionalumo tyrimas ....	48

## PAVEIKSLŲ SĄRAŠAS

1.1 pav. E-mokymo prieinamumas .....	9
1.2 pav. Pedagogo vaidmuo ir jo veikla e-mokyme.....	16
1.3 pav. Elektroninio kurso komponavimo proceso modelis (1 žingsnis) .....	16
1.4 pav. Detalizuotas proceso modelis .....	17
1.5 pav. SCORM standarto kūrėjai .....	21
1.6 pav. Turinio paketo struktūra .....	23
1.7 pav. Manifesto failo struktūros pavyzdys .....	23
1.8 pav. Elektroninio mokymo bendra funkcinė schema .....	24
1.9 pav. Elektroninio kurso atsiradimo kelias .....	25
1.10 pav. Elektroninio kurso kūrimo ir komponavimo schema .....	25
1.11 pav. Kursų mokymo objektų komponavimo sistemos architektūra .....	26
1.12 pav. Elektroninių kursų mokymo objektų komponavimo sistemos panaudojimo atvejai.....	27
2.1 pav. Komponavimo proceso detali panaudojimo atvejų schema .....	32
2.2 pav. Mokymo objekto komponavimo sekos diagrama.....	34
2.3 pav. Metaduomenų įkėlimas .....	34
2.4 pav. Metaduomenų aprašymas sekų diagrama .....	35

2.5 pav. Turinio pakavimo sekų diagrama .....	35
2.6 pav. Elektroninių kursų mokymo objektų organizavimas.....	36
2.7 pav. Mokymo objekto atvaizdavimas MVS .....	37
2.8 pav. Mokymo objekto atvaizdavimas naršyklėje .....	37
2.9 pav. Pradinė kursų kūrimo architektūra .....	39
2.10 pav. Elektroninių kursų mokymo objektų komponavimo sistemos komponentinė schema; .....	40
2.11 pav. Turinio pakavimo sistemos išsami klasių diagrama .....	41
2.12 pav. Meta-duomenų posistemės išsami struktūra.....	42
2.13 pav. MD_Aprasymas posistemės išsami struktūra.....	43
2.14 pav. Realizacijos modelis .....	44
2.15 pav. Vartotojo sąsajos projektas, parinis vedlio langas.....	45
2.16 pav. Vartotojo sąsajos projektas metaduomenų aprašymui .....	46
3.1 pav. dokebosLMS2004 kurso įkėlimo langas .....	50
3.2 pav. dokebosLMS2004 sistemoje įkeltas eksperimentinis kursas .....	50
3.3 pav. ATutor 1.5.2 kurso įkėlimo langas .....	50
3.4 pav. ATutor 1.5.2. pranešimas po kurso įkėlimo .....	51
3.5 pav. dokebos 1.6.3 kurso įkėlimo langas .....	51
3.6 pav. dokebosLMS2004 pagrindinio puslapio atvaizdavimas.....	52
3.7 pav. dokebosLMS2004 turinio vaizdavimas.....	52
3.8 pav. ATutor pagrindinio puslapio atvaizdavimas .....	53
3.9 pav. ATutor turinio vaizdavimas.....	53
3.10 pav. Dokeos pagrindinio puslapio atvaizdavimas .....	54
3.11 pav. Dokeos turinio vaizdavimas .....	54
4.1 pav. 1 žingsnis: Apie programą;.....	57
4.2 pav. 2 žingsnis: mokymo objekto išrinkimas .....	57
4.3 pav. Mokymo objekto vietos nurodymo langas ir galimi filtrai.....	58
4.4 pav. Vartotojo perspėjimas, jei toks katalogas jau egzistuoja.....	58
4.5 pav. 3 žingsnis: Mokymo objekto atvaizdavimas .....	58
4.6 pav. 4 žingsnis: Mokymo objekto meta-duomenys.....	59
4.7 pav. 5 žingsnis: Kuriamas SCORM paketas .....	59
4.8 pav. 6 žingsnis: Baigiamas žingsnis.....	60
4.9 pav. Pagrindinių <general> metaduomenų aprašymas.....	60
4.10 pav. Gyvavimo ciklo <lifeCycle> aprašymas .....	61
4.11 pav. <contribution> įrašo pridėjimas .....	61
4.12 pav. Techninių <technical> duomenų aprašymas .....	62
4.13 pav. Mokymo objekto apribojimų <rights> aprašymas .....	62
4.14 pav. Klasifikatoriaus <classification> aprašymas .....	63
4.15 pav. Klasifikatoriaus įrašų redagavimas.....	63

## ĮVADAS

Viena pagrindinių žmonijos vystymosi sąlygų – sukauptų žinių, patyrimų perdavimas ateities kartoms. Sukūrus geras žinių perdavimo metodologijas, atsiranda tų žinių pasiekiamumo, prieinamumo problema. Pastaruoju metu tobulėjant informacinėms technologijoms, susidaro vis palankesnės sąlygos nuotoliniam mokymui priartėti prie tradicinio mokymo. O tai sudaro sąlygas paimti iš abiejų visa, kas geriausia, ir sukurti elektroninio mokymo sistemą, galinčią teikti visavertį išsilavinimą, reikalingą pilnaverčiam funkcionavimui informacinėje visuomenėje. Tai labai svarbu norint įgyvendinti elektroninės erdvės (el. valdžia, el. mokykla) priartinimą prie paprastų vartotojų. Informacinės visuomenės kūrimas gali tapti aiškiu visos švietimo sistemos ir kiekvienos mokyklos vystymo stimulu.

Atkreipsime dėmesį į tai, kad „Nacionalinėje informacinės visuomenės plėtros koncepcijoje“ (2001-02-28, nutarimo Nr. 229) pirmiausia pabrėžiama informuotumo ir kompetencijos svarba. Dokumente suformuluoti informacinės visuomenės kūrimo Lietuvoje uždaviniai, paminėsime keletą svarbesnių mūsų darbui:

- sudaryti sąlygas, kad **kiekvienas moksleivis mokykloje įgytų reikiamas informacinėmis technologijomis grindžiamas žinias;**
- suteikti galimybę studentams, dėstytojams, mokslo, kultūros darbuotojams naudotis informacinėmis technologijomis;
- **sukurti visuomenei lengvai pasiekiamą nuotolinių studijų sistemą, paremtą informacinių technologijų naudojimu;**
- pasiekti, kad kiekvienas mokyklą baigęs moksleivis įgytų reikiamas informacinėmis technologijomis grindžiamas žinias;
- sukurti **atvirus visuomenei prieigos taškus**, ypač kaime, per kuriuos gyventojai galėtų kaupti informacinių technologijų naudojimo patirtį.<sup>1</sup>

Informacinių technologijų (IT) vystymasis lemia ir tai, kad joms skiriama vis daugiau dėmesio vidurinėse mokyklose (pvz., „Nacionalinę informacinės visuomenės plėtros koncepcija“). Todėl labai svarbu tampa mokėti naudotis, dirbti, pritaikyti įvairias kompiuterines sistemas. Tuo pačiu pats IT mokymo procesas tampa labai svarbia ir kitų dalykų mokymo dalimi. Kadangi šiuolaikinėje visuomenėje viskas vis greičiau kinta, atsiranda poreikis tokiai mokymo sistemai, kuri būtų lengvai valdoma, perprantama, o svarbiausia, lengvai adaptuojama keičiantis mokomo dalyko turiniui ar aplinkai (ne tik informacinių technologijų atveju). Pagrindiniu vertinimu tampa ne tai, ką žinai, moki, bet per kiek laiko *gali* tai sužinoti, išmokti. Pereinama prie nuolatinio mokymosi, kuris leidžia kelti kvalifikaciją neatitraukiant nuo darbų. Todėl atsiranda poreikis mokymo sistemai, suteikiančiai tokias galimybes.

<sup>1</sup> Valstybės Žinios: Lietuvos nacionalinė informacinės visuomenės plėtros koncepcija. Vilnius, 2001, Nr. 20-652.

**Pagrindinis nuotolinių studijų per internetą pranašumas – „studijų iš bet kurios vietos ir bet kuriuo laiku” galimybė.** Dažnai pažymimas ir galimas nemokamų studijų aspektas, nors kol kas dauguma elektroninių studijų sistemų yra komercinės, o daugelis studijų institucijų siūlo mokamas nuotolinių studijų programas. Pastarųjų metų ryšio priemonių išplitimas ryškiai keičia ir informacijos pateikimo būdus: vaizdai internete tapo įprasti, vis intensyviau naudojami garso ir vaizdo įrašai, pasirodo ir kiti įvairialypės terpės tipai. Tai leidžia mokymo medžiagą pateikti efektyviau ir kokybiškiau, įvesti objektyvesnio vertinimo sistemas, realizuoti pilnavertį grįžtamąjį ryšį (*feedback*). „Naujausi elektroniniai studijų kursai perkeliama iš modelio „paskaitos ir galutinis egzaminas arba vienas didelis projektas” į modelį su penkiomis-septyniomis tarpinėmis kontrolinėmis užduotimis arba didelio projekto dalimis, kurių kiekviena paprastai pateikiama per žiniatinklio pagrindu sukurtą elektroninių studijų sistemą, ir kiekviena užduotis vertinama tam tikra viso kurso balo dalimi.“<sup>2</sup> Problema tampa operatyvus, greitas, objektyvus kurso valdymas.

Didėjant elektroninio mokymo sistemų įvairovei tampa vis sunkiau sudaryti kursus, ypač jų dalis, kurias po tam tikro laiko būtų galima panaudoti skirtingose sistemose nekeičiant struktūros, aprašymų, duomenų. Kiekviena sistema stengiasi savaip pateikti kursus, todėl kursų kūrėjui reikia taikytis prie sistemos reikalavimų, o tai mažina laiką bei koncentraciją pačiam kurso turinio kūrimui. Be to, tokius kursus, kurtus konkrečiai aplinkai, sunku panaudoti kitose sistemose neįdedant daug darbo, atsiranda papildomi kursų kaštai. Kaip ir kitose srityse, panašias problemas galėtų išspręsti standartai.

Pagrindinis šio darbo tikslas – sukurti įrankį, kuris leistų kurti elektroninio mokymo kursus nepriklausomai nuo mokymo valdymo sistemos (MVS). Kursas turi būti aprašytas taip, kad jį būtų galima lengvai keisti, panaudoti jo dalis kituose kursuose (sukomponuotas), todėl sistema turi dirbti mokymo objektų lygmenyje. Tokiu būdu kursas gali būti pateiktas per bet kurią elektroninio mokymo sistemą neprarandant kurso raiškos, funkcionalumo, efektyvumo ir vaizdumo, - taigi kursas tampa pilnaverte preke.

Tikslo įgyvendinimui buvo atlikti šie darbai:

- analitinis darbas identifikuojant problemą;
- detalizuotas ir schematizuotas komponavimo procesas;
- sudarytas sistemos paketinis projektas;
- sukurtas vartotojo sąsajos modelis;
- realizuotas elektroninių kursų mokymo objektų komponavimo įrankis;
- atliktas eksperimentinis sukurtos sistemos tyrimas;
- atliktas eksperimentinis sistemos duodamų rezultatų teisingumo tyrimas;
- sudarytas realizuotos sistemos vartotojo vadovas.

<sup>2</sup> Ambrazevičius E., Jasiukevičius A., Šakys V. Universitetinių e.studijų sistemų kūrimo principai ir problemos. [žiūrėta 2004-11-19]. Prieiga per internetą: <http://www.leidykla.vu.lt/inetleid/inf-mok/22/str3.html>



# 1. ELEKTRONINIŲ KURSŲ KŪRIMO SISTEMŲ ANALIZĖ

Atliksime analitinį darbą, tiksliai apibrėšime darbo tikslą ir tyrimo sritis. Apžvelgsime elektroninio mokymo teorijos pagrindus, apibrėšime būsimus vartotojus, nustatysime būsimus sistemos kriterijus, parinksime elektroninių kursų kūrimo sistemos projektavimo ir realizavimo įrankius.

## 1.1. SISTEMAI KELIAMSI REIKALAVIMAI BEI NUMATOMAS FUNKCIONALUMAS

Pagrindinis šio darbo tikslas yra sukurti patikimą ir lanksčią elektroninių mokymo kursų kūrimo sistemą. Kurso kūrimo sistema turi būti *nepriklausoma* nuo medžiagos pateikimo aplinkų, lengvai pritaikoma. Apibrėšime, kokius reikalavimus turėtų tenkinti būsima sistema, suskirstysime juos į dvi dalis:

- Sistemos ir kurso poreikiai:
  - *adaptyvumas* – lengvai pritaikomi mokymo turinio pateikimo sistemoms;
  - *prieinamumas* – leidžia paimti komponentus iš vienos ir perkelti į kitas vietas ar net kursus, suteikia galimybę panaudoti mokymo objektų viešas saugyklas (pav. 1.1);



1.1 pav. E-mokymo prieinamumas

- *patogumas* – lengvas administravimas, pertvarkymas;
- *suderinamumas* – kiek įmanoma didesnis įvairių platformų palaikymas;
- vartotojų poreikiai:
  - *patrauklumas* – lanksti ir patogi vartotojo sąsaja;
  - *patogumas* – lengvai perprantama ir informatyvi vartotojo sąsaja;
  - *palanki mokymosi aplinka* – ne tik perduoti informaciją bei žinias, bet ir kurti mokymui palankią mokymosi aplinką. **Mokomasi tikrai tada, kai yra konkreti skatinimo sistema ir kai siekiama gauti konkrečius rezultatus.**
  - *prieinamumas* – galimybė naudotis sistema bet kuriuo laiku, bet kurioje vietoje ir kuo pigiau (pvz., sistemos Web versija).

Norint sukurti tokią elektroninio kursų komponavimo sistemą, pirma reiktų panagrinėti nuotolinio mokymosi pagrindus, atlikti elektroninių studijų sistemų kūrimo principų ir problemų analizę, api-

brėžti grįžtamojo ryšio organizavimo principus, išsiaiškinti ir susipažinti su jau veikiančiomis kitomis elektroninio mokymo sistemomis bei kursų kūrimo įrankiais. Šiam tikslui panagrinėsime keletą sukurtų sistemų, apžvelgsime jų privalumus ir trūkumus tiek Lietuvoje, tiek užsienio šalyse. Galiausiai turime pasirinkti elektroninio mokymo sistemos projektavimui bei realizavimui reikalingas priemones.

## 1.2. PROBLEMOS SPRENDIMŲ ANALIZĖ

Pirmiausia, norint įsigilinti į būsimos sistemos kūrimo niuansus panagrinėsime literatūros šaltinius ir apibrėšime, kokios žinios mums reikalingos.

Kuriant elektroninio mokymo kursų komponavimo sistemą turėtų būti nagrinėjamos šios esminės sritys - tai pačios komponavimo sistemos kūrimo *ypatumai* bei vartotojų *poreikiai* – mūsų atveju didžiausias dėmesys skirtas *kursų kūrimui* ir *aprašymui*.

Norint sukurti vertingą sistemą, reikia suprasti pagrindinius elektroninio mokymo principus, pačių sistemų veikimą, kursų pateikimą, susipažinti su susijusiomis technologijomis. Būtina gerai išsiaiškinti jau egzistuojančių sistemų privalumus bei trūkumus, surasti pagrindines problemines sritis, kurios šiuo metu egzistuoja realizuojant bei naudojant elektroninių kursų kūrimo sistemas.

Formaliai apibrėšime dominančias elektroninio mokymo sritis, kurios būtinos norint sukurti funkcionalią elektroninių kursų komponavimo sistemą:

- Mokymo turinio valdymo sistemos (MTVS);
- Mokymosi valdymo sistemos (MVS);
- Mokymo metodai elektroniniame mokyme;
- Mokymo medžiagos ruošimas elektroniniam mokymui;
- Dabar egzistuojančių elektroninių kursų kūrimo sistemų sprendimai.

**Tyrimo sritis:** elektroninis mokymas.

**Objektas:** elektroninio mokymo kursų komponavimo sistema. Sukurta sistema ir sistemos duodamas rezultatas - kursas turi tenkinti visus anksčius šiame skyriuje ir įvade apibrėžtus reikalavimus.

## 1.3. ELEKTRONINIO MOKYMO PAGRINDAI

Teoriškai elektroninį mokymą galima suskaidyti į tokius lygius:

- elektroninė knyga;
- elektroninis kursas;
- elektroninė sistema;
- elektroninė mokykla (universitetas).

Šiuolaikinėje informacinėje visuomenėje, kurioje svarbiausiomis vertybėmis tampa informacija ir naujos technologijos, vis daugiau dėmesio skiriama nuotolinio mokymosi plėtrai, kurio dalyviams suteikiama galimybė mokytis jiems patogiu metu ir priimtina sparta pasirinktoje vietoje. Elektroninės

studijos suteikia galimybę mokytis ir tobulėti visą gyvenimą, įgalina lengviau pritaikyti ir įsitvirtinti nuolat kintančioje ir kaskart daugiau reikalavimų keliančioje darbo rinkoje.

Tobulėjančios informacinės ir komunikacinės technologijos daro įtaką ir nuotolinio švietimo plėtočiai. Nuotolinės studijos perkeliamos į naują – virtualų - lygmenį, atskleidžiantį naujas studijų organizavimo galimybes.

Nuotolinių studijų ypatumai:

- *Lankstumas* - galimybė mokytis patogiu laiku, patogioje vietoje, norimu tempu.
- *Modulinė struktūra* - galimybė iš atskirų kursų modulių sudaryti individualius ar grupinius mokymosi planus.
- *Technologiškumas* - naujausių informacinių ir komunikacinių technologijų panaudojimas (multimedija).
- *Naujas dėstytojo vaidmuo* - atsiranda naujos funkcijos: mokymosi proceso koordinavimas, konsultavimas, dėstomo kurso koregavimas, vadovavimas projektams ir pan.
- *Nauji reikalavimai besimokančiajam* - būtina tiksli motyvacija, savarankiškumas, darbštumas, tam tikri pagrindiniai technologijų naudojimo įgūdžiai.
- *Dvipusė komunikacija* - studentai ir dėstytojai bendrauja tiek sinchroniškai, tiek asinchroniškai.
- *Lygiagretumas* - galimybė studijuoti, neatitrūkstant nuo profesinės veiklos, iš karto pritaikant naujai įgytus įgūdžius praktikoje.
- *Ekonomiškumas* - užsienyje atliktų tyrimų duomenys rodo, kad nuotolinės studijos yra pigesnės nei tradicinės.
- *Socialinė lygybė* - vienodos galimybės įgauti išsilavinimą, nepriklausomai nuo gyvenamosios vietos, sveikatos būklės ar materialinių išteklių.
- *Tarptautiškumas* - pasaulinių laimėjimų švietimo paslaugų rinkoje eksportas ir importas, tarptautinis bendradarbiavimas.<sup>3</sup>

#### **1.4. ELEKTRONINIS MOKYMAS LIETUVOJE IR UŽSIENYJE**

Šiuo metu Lietuvoje egzistuojančios elektroninio mokymo sistemos vystosi, plečiasi, daugėja įvairių nuotolinio mokymo kursų. Numatomos steigti distancinio mokymo klasės. Labiausiai išvystytas kompiuterinio raštingumo mokymas, siūlomas daugeliu įvairių firmų.

Elektroninės mokyklos užuomazgų jau galima išvelgti, bet iki pilnaverčių mokymo institucijų dar toli. Nors jau egzistuoja plačiai paplitę tradiciniai, mokyklai skirtų IT temų mokymo būdai. Pavyz-

<sup>3</sup> Rutkauskienė D. Įvadas į distancines studijas: paskaita. [žiūrėta 2004-12-10]. Prieiga per internetą: <http://distance.ktu.lt/vips/sessionrecords.php>.

džiui, Vilniaus "Ozo" vidurinėje mokykloje 2003m. įsteigta 10-a, o nuo 2004m. rugsėjo mėn.1 d. įsteigta 11-a nuotolinio mokymo klasė.\*

Užsienyje šiuo atžvilgiu yra toliau nueita, ten jau dabar egzistuoja elektroninės mokyklos ir net elektroniniai universitetai, teikiantys visapusišką ir pilnavertį išsilavinimą. Įvairių kursų gausybė gali klaidinti, jų yra daug ir įvairių. Aptariamuoju atveju mes domėjomės tik informacinių technologijų mokančiomis sistemomis. Pateikiamus kursus galima skirstyti pagal tai, ar jie mokami, ar nemokami. Nemokami dažniausiai būna paprastesni, supažindinantys su tam tikrais pagrindais, kursai. Pvz., <http://elearning.asu.edu/itf/> moko IT pagrindų, svarbiausių sąvokų.

Mokami kursai dažai būna profesionalesni, geriau paruošti. Sunku įvertinti jų kokybę, jų pačių neišbandžius. Pavyzdžiui:

- <http://www.un.int/unitar/patit/globaltraining/onlinelearningprogram.htm> - tai Jungtinių Tautų institutas, teikiantis IT kursus, tokius kaip elektroninio pašto pagrindai, paieška internete ir t.t.
- <http://www.solartechnj.com/course/sear.htm> - daug įvairiausių kursų nuo interneto, teksto rengyklų iki programavimo, skirtų pradedantiesiems ir ne tik.

Lietuvoje kitokia situacija. Žinoma, rinkoje yra keletas modulių, tinkančių ir mokiniams. Pavyzdžiui, „Straujos kompiuterijos mokymo centras“ (<http://www.itmc.lt/>) skirtas ir moksleiviams, bet labiau orientuojamas į ECLD (European Computer Driving Licence) programą (kompiuterinio raštingumo standartu), trūksta įvairovės. Vis dėlto tokie kursai savo pobūdžiu yra skirti žmonėms, norintiems įgyti kompiuterinių įgūdžių, nekeliant didesnių uždavinių ir reikalavimų, todėl netinkami toliau pažengusiems šioje srityje.

## **1.5. ELEKTRONINIŲ MOKYMO SISTEMŲ KŪRIMO PRINCIPŲ IR PROBLEMŲ ANALIZĖ**

Elektroninių mokymo sistemų kūrimo principų ir problemų analizė atlikta remiantis naujai interpretuojamomis koncepcijomis, poreikiais bei keliamais reikalavimais.

Norima sukurti prieinamą elektroninio mokymo kursą besimokantiesiems ne auditorijose ir kartu turėti patrauklią, patį besimokantįjį vertinančią mokymosi aplinką. Mokymosi aplinkoje taip pat turėtų būti ugdomi ir besimokančiųjų bendrieji įgūdžiai:

- informacinis ir kompiuterinis raštingumas;
- gebėjimas studijuoti savarankiškai;
- problemų sprendimas;
- kritinis mąstymas;
- asmeniniai ir grupiniai gebėjimai, reikalingi bendravimui;

---

\* [http://www.vilnius.lt/new/gidas.php?open=45&root=1&sub\\_cat1=395&id=646](http://www.vilnius.lt/new/gidas.php?open=45&root=1&sub_cat1=395&id=646)

- kolektyvinis darbas su komanda;
- vadovavimas komandai.

Be to, turi būti kuriama elektroninių studijų aplinka, kai naudojamos šiuolaikinės įvairialypės terpės (multimedijos) priemonės ir komunikavimo technologijos, kurios užtikrintų būtinus dėstytojų (mokytojų) ir mokinių (besimokančiojo) ryšius mokymosi metu.

Technologiniai pokyčiai veikia visuomenę, ir kiekvienas jos narys turi nuolat įgyti naujų žinių. Vis dažniau pabrėžiama nuomonė, kad vieną kartą įgyto bei tinkamo visam gyvenimui išsilavinimo laikai jau praėjo. Galimybės studijuoti visą gyvenimą turi būti suteikiamos ir įvairių rūšių veikla užsiimantiems žmonėms, ir nereguliariai (kartkartėmis) besimokantiems specialistams, be to, jie dar gali turėti ir skirtingą bazinį išsilavinimą, ir skirtingus žinių lygmenis ar studijų stilius<sup>4</sup>.

Daugiau ar mažiau išsamūs apklausos ir testavimo metodai pateikia sistemai tikslesnę informaciją apie studentą, - tai didelis jų pranašumas. Vertinama ir tai, kad vartotojo modelio kūrimas yra permatomas (skaidrus) studentui, o šis gali jį tiesiogiai veikti.

Kadangi dėl realaus žmogaus pasaulio sudėtingumo vartotojo modeliavimo psichologinis pagrindas dar nėra aiškiai apibrėžtas, todėl siekiama leisti vartotojui pačiam dalyvauti modeliuojant save, o vėliau koreguoti galimus neteisingus interpretavimus. Tačiau rinkdamas šią išsamią informaciją vartotojas perkraunamas. Priverstas išmokti naudotis elektroninio mokymo programine įranga ir atlikti normalaus studijų proceso veiksmus, vartotojas dar turi sugaišti nemažai laiko nurodydamas sistemai, kas jis yra ir ko nori. Todėl rekomenduojama nepersistengti klausinėjant ir testuojant mokinius.

Kita vertus, nustatytas vartotojo studijų kelias taip pat turi pranašumų, nes jis leidžia vartotojui iš karto susikaupti dalyko nagrinėjimui ir netrukdo jo daugeliu klausimų. Tačiau šiuo atveju, nenaudojant išsamios vartotojo grįžtamojo ryšio sistemos, studijų procesas gali būti suboptimalus ir net neadekvačiai adaptuotas. Todėl reikėtų siekti aptartus studijų metodus suderinti taip, kad jie leistų pasirinkti optimalią strategiją.

## 1.6. GRĮŽTAMASIS RYŠYS

Pastaruoju metu darbuose, tyrinėjančiuose elektroninio mokymo sistemas, vis dažniau akcentuojamas tarpinių kontrolinių užduočių (assignment) ir grįžtamojo ryšio (feedback) vaidmuo. Naujausi elektroninio mokymo kursai perkeliama iš modelio „paskaitos ir galutinis egzaminas arba vienas didelis projektas“ į modelį su keliomis tarpinėmis kontrolinėmis užduotimis arba didelio projekto dalimis, kurių kiekviena paprastai pateikiama per žiniatinklio pagrindu sukurtą elektroninių studijų sistemą, ir kiekviena užduotis vertinama tam tikra viso kurso balo dalimi.<sup>5</sup>

<sup>4</sup> Aušra A. Mokslinė elektroninė biblioteka kaip e-studijų priemonė. [žiūrėta 2005-08-10]. Prieiga per internetą: [http://www.elibrary.lt/ktu\\_konf\\_pranes.phtml](http://www.elibrary.lt/ktu_konf_pranes.phtml);

<sup>5</sup> Ambrazevičius E., Jasiukevičius A., Šakys V. Universitetinių e.studijų sistemų kūrimo principai ir problemos. [žiūrėta 2005-05-09]. Prieiga per internetą: <http://www.leidykla.vu.lt/inetleid/inf-mok/22/str3.html>

Svarbu mokymosi metu pastoviai palaikyti grįžtamąjį ryšį su besimokančiais ir patiems reaguoti į besimokančiųjų veiklą bei pastangas. Grįžtamasis ryšys gali būti palaikomas *apklausomis, paklausimais elektroniniu paštu ar diskusijų forumais*. Tai turėtų besimokantiejiems padėti užtvirtinti jų supratimą ir jį tobulinti.<sup>6</sup>

Besimokančiųjų veiklos sritys gali būti pateiktos keliais būdais. Pavyzdžiui:

- Savikontrolės klausimai ir pratimai įtraukti į mokymosi medžiagos turinį.
- Savikontrolės testas skyriaus pabaigoje, kuris patikrina, ką besimokantysis išmoko iš viso skyriaus.
- Užduotys ir pratimai, kurių reikėtų imtis po daugelio darbo valandų įtraukiant ekstensyvią praktiką ar projektinį bei grupinį darbą.<sup>7</sup>

Besimokantysis gali pateikti atsakymus į užklausas ar užduotis įvairiais būdais:

- apmąstyti atsakymą, bet jo niekur nepateikti;
- pažymėti langeliuose teisingus atsakymus;
- nuspausti kompiuterio klavišus;
- užpildyti lentelę;
- užpildyti tuščias užklausų vietas;
- įrašyti laukeliuose žodžius, frazes ar skaičius;
- nubrėžti diagramą ar grafiką;
- atlikti skaičiavimus ir t.t.<sup>8</sup>

Žiniatinklio grįžtamasis ryšys leidžia naudoti skirtingas terpes: teksto forma pateikiamus elektroninius dokumentus (elektroninį paštą, pokalbių svetaines, formų užpildymą, lenteles arba pridėtus dokumentus). Grįžtamąjį ryšį galima pateikti ir naudojant įvairialypės terpės priemones.

## 1.7. ELEKTRONINIO MOKYMO MEDŽIAGOS RUOŠIMAS

Elektroninio mokymosi medžiaga ruošiama pagal specifinius reikalavimus. Kursas susideda iš modulių. Moduliai skirstomi į skyrius ir poskyrius.

Atvirojo mokymosi medžiagoje būtina numatyti besimokančiųjų problemas, studentų poreikius, įvairinti mokymo metodus bei įvertinti mokymuisi skiriamą laiką. Būtina griežtai atskirti specialiąsias žinias, kurių reikia rengiamam kursui, nuo bendrųjų žinių, t. y. dėstomo dalyko pagrindų. Galima improvizuoti kaip ir tradicinėje paskaitoje, tačiau reikia atsiminti, kad asmenys, kurie mokosi nuotoliniu būdu, negali matyti veido išraiškos ir sakiniui gali suteikti kitokią prasminę atspalvį nei buvo ketinta perteikti.

<sup>6</sup> Rutkauskienė D. Nuotolinis mokymasis (Kovertaitė V.R. Mokomosios medžiagos rengimas.). Kaunas: Technologija, 2003. p.83.

<sup>7</sup> Ten pat.

<sup>8</sup> Ten pat.

Savikontrolės klausimai leidžia patikrinti numatytus mokymosi tikslus. Vienas savikontrolės klausimas skiriamas vienam mokymosi tikslui, tačiau juo galima patikrinti ir keletą mokymosi tikslų. Savikontrolės klausimas įvertina studento mokėjimus ir gebėjimus bei pakeičia medžiagos suvokimo spartą bei pobūdį.

Atvirojo mokymosi medžiagoje dažnai pateikiami tik keli puslapiai teksto be jokios praktinės užduoties. Vidutiniškai vienam puslapiui turėtų tekti viena praktinė užduotis.

Nuotolinių studijų kurse turi būti naudojamos daugialypės terpės (paveikslai ir pagal poreikius bei galimybes vaizdo ar garso įrašai arba mokomosios imitacinės programos). Be to, šios priemonės gali būti tiesiogiai nesusijusios su kursu ir turi leisti studentui prasiblaškyti nuo monotoniško skaitymo.

Rengiant distancinio mokymosi medžiagą reikia prisiminti, kad „svarbiausias mūsų tikslas - padėti studijuojantiems įgyti reikalingų žinių ir įgūdžių!”.<sup>9</sup>

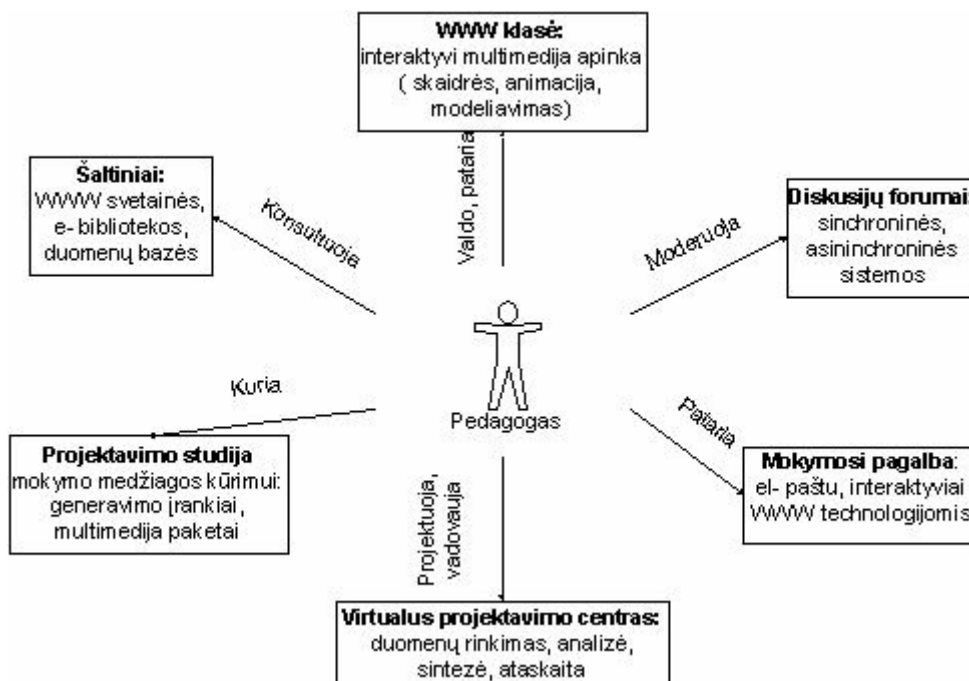
## **1.8. PEDAGOGO (MOKYTOJO) VEIKLA ELEKTRONINIAME MOKYME**

Elektroninis mokymas - tai žinių pateikimo būdas, o ne specialybės aspektas. Naudojant IT plečiasi pedagogo veiklos sritys mokymo procese. Taikomuojų ir funkcinį požiūriu mokytojo veikla yra tokio pobūdžio (1.2 pav.):

- Vadybinė - registruoja studentus, planuoja mokymo procesą, tvarkaraštį, organizuoja mokymo įvertinimą.
- Profesinė - kelia savo kvalifikaciją, bendrauja su kolegomis.
- Technologinė (komercinė) - ruošia ir publikuoja mokymo medžiagą.
- Pedagoginė - instruktuoja, pataria, valdo studentų žinių įsisavinimą, vertina jų darbus ir žinias.

---

<sup>9</sup> Rutkauskienė D. Nuotolinis mokymasis (Abarius P., Mačiulis M. Virtuali mokymosi terpė WebCT). Kaunas: Technologija, 2003. p.120.



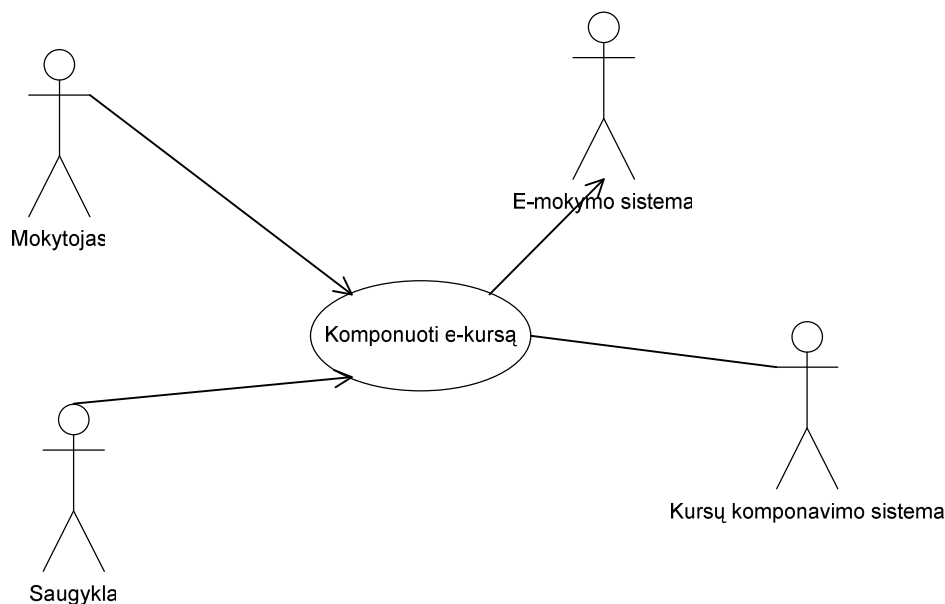
1.2 pav. Pedagogo vaidmuo ir jo veikla e-mokyme

## 1.9. ELEKTRONINIŲ KURSŲ KŪRIMO APLINKOS ANALIZĖ

Nustatę pradinis teorinius reikalavimus, galime formaliai nustatyti, kokia turėtų būti kursų kūrimo sistema.

Atliksime pirminę aplinkos analizę. Tam pasirinkome kurso komponavimo procesą, nes tai yra esminė būsimos sistemos funkcija – *komponuoti* elektroninio mokymo kursas.

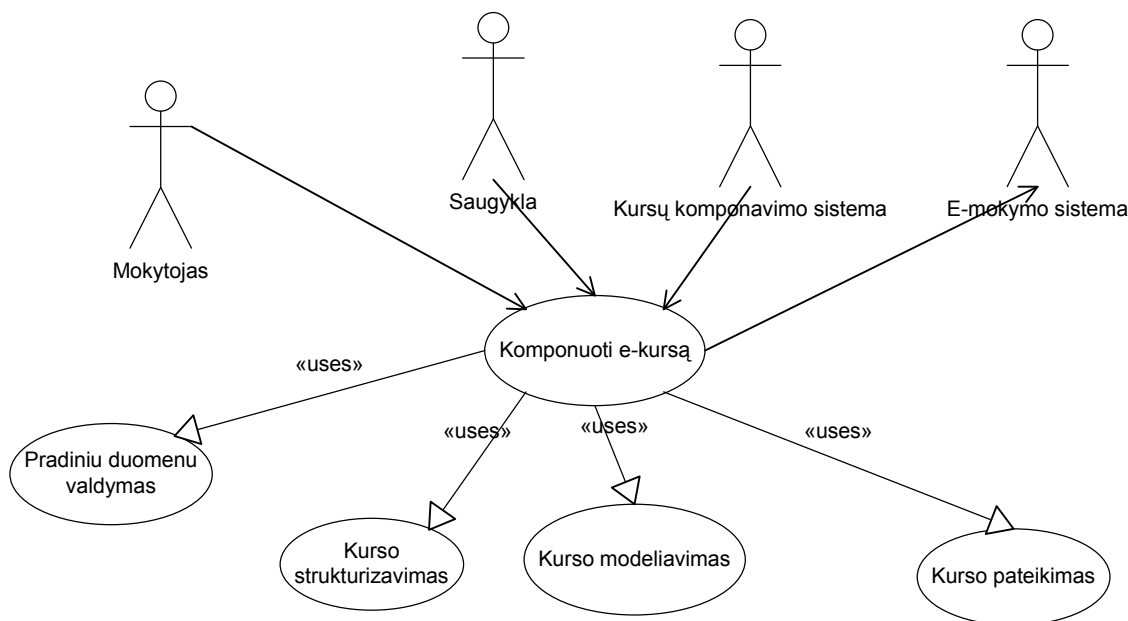
1.3 paveiksle parodytas pradinis proceso „Komponuoti elektroninį kursą“ modelis:



1.3 pav. Elektroninio kurso komponavimo proceso modelis (1 žingsnis)

Kadangi mes iš pirminio modelio matome tik proceso dalyvius, sudarome detalizuotą modelį (1.4 pav.), kuriuo jau galime apibrėžti smulkesnius procesus.





1.4 pav. Detalizuotas proceso modelis

Toliau nagrinėjant būtų galima detalizuoti modelį iki smulkesnių procesų. Tai bus atlikta vėliau nustatant sistemos reikalavimus.

### 1.9.1. VARTOTOJŲ TIKSLAI IR PROBLEMOS

Apžvelgsime numatomų vartotojų tikslus, poreikius ir problemas.

Vartotojų tikslai:

- greitesnis ir aiškesnis kursų kūrimas;
- kursai, nepriklausantys nuo pateikimo terpės (elektroninės mokymo sistemos);
- didesnis funkcionalumas.

Paslaugos tiekėjų tikslai:

- mokymo kokybės gerinimas;
- vertinimo objektyvumo (aiškumo) gerinimas;
- mokslo prieinamumo didinimas;
- kursų kūrimo standartizavimas.

Vartotojų problemos:

- kurso efektyvumo patikrinimo problemos;
- lengvas suderinamumas ir integracija.

### 1.9.2. ELEKTRONINIO MOKYMO SISTEMOS BRUOŽAI

Nuotolinio mokymo teorijoje nurodomi šie pagrindiniai kriterijai, reikalingi pilnavertei elektronei mokymo sistemai:

- išmokti suprasti, kaip technologijos keičia mokymą, planuoti nuolatinį vystymąsi;
- įvertinti mokymo modulius, jų tinkamumą, jų dėstymo lygį;

- mokyti personalą naudoti IT modulių ruošimui, mokymo medžiagos parengimui, mokymo procese ir valdyme;
- mokyti elektroninio mokymo principų ir metodikos, suinteresuoti sprendimus priimančią vadovybę;
- kurti reikalingas interaktyvias elektroninio mokymo sistemas, tyrinėti mokymo modelius;
- bendradarbiauti su užsienio universitetais ir jungtis į mokymo bendrijas;
- reikia nuolat atnaujinti kompiuterinę ir programinę įrangą;
- reikia įtraukti į mokymo planavimą IT personalą.

### **1.9.3. ELEKTRONINIO MOKYMO KURSŲ KŪRIMO SISTEMOS**

Šiame skyriuje pabandysime apžvelgti praktines kursų kūrimo sistemų realizacijas.

Pasaulyje egzistuoja gana nemažai mokymo kurso kūrimo aplinkų. Viena iš geriau žinomų - elektroninio mokymo kurso rengimo aplinka CDK. Kursų kūrimo aplinka CDK yra skirta elektroninei medžiagai – elektroninėms knygoms ir nuotolinio mokymosi studijų kursams – kurti. Aplinka yra pasiekama internetu ir pirmiausia orientuota į mažai IKT žinių turinčius vartotojus. Tai patogi, lengvai valdoma ir suprantama aplinka tiems vartotojams, kurie turi darbo įgūdžių su teksto redaktoriais (pvz., MS Word). CDK aplinka sukurtus kursus šiuo metu galima eksportuoti į HTML puslapius, tinkančius internetui ir virtualioms mokymosi aplinkoms (pvz., WebCT), teikiančioms mokymosi kursus interneto tinklapiu forma. CourseML kalba kurso medžiaga, aprašoma struktūriškai, todėl lengvai gali būti redaguojama ir naudojama pakartotinai. Be to, tik struktūriškai aprašyta elektroninių kursų medžiaga ateityje gali būti suderinta su kuriamais standartais EMK (elektroniniams mokymosi kursams) aprašyti. Pati kurso medžiaga gali būti rašoma įvairiomis kompiuteriui pažįstamomis kalbomis.<sup>10</sup>

Pagrindinė visų tokių kursų kūrimo sistemų problema – standarto nebuvimas. Dažniausiai kursas kuriamas jau žinant, kokioje terpėje jis bus pateikiamas. Tai labai mažina kurso kūrėjų galimybes ir medžiagos pateikimo efektyvumą. Be to, keičiantis elektroninio mokymo sistemai, reikia gan nemažai pakeitimų ir pačiame mokymo kurse. Standarto nebuvimo problema susijusi ne tik su kursų kūrimu, bet ir su visu elektroniniu mokymu. Anksčiau apibrėžtam vartotojo poreikiui sistemos nepriklausomumui nuo pateikimo terpės būtinas standarto įvedimas.

### **1.9.4. DABARTINIAI ELEKTRONINIO MOKYMO STANDARTAI**

Šiame skyrelyje pabandysime apžvelgti šiuo metu egzistuojančius elektroninio mokymo standartus ir ruošiamas priimti specifikacijas.

- AICC – koncentruojasi ties mokymo turinio suderinamumu su kuo daugiau elektroninių mokymo sistemų.

<sup>10</sup> Rutkauskienė D. Nuotolinis mokymasis (Kubiliūnas R., Cibulskis G. E. mokymosi kurso aplinka CDK). Kaunas: Technologija, 2003. P.186.

- IEEE – pagrindinis tikslas specifikuoti mokymo objektų meta-duomenis.
- IMS – užsiima XML struktūrų specifikuojimu. Šiuo atveju kursų, turinio, meta-duomenų aprašymu XML.
- SCORM – tai ADL kompanijos kuriamas standartas, kurio pagrindiniai tikslai yra užtikrinti lengvą mokymo priemonių priėjimą, kūrimą, naudojimą kuriant ir naudojant elektroninius kursus. SCORM remiasi IEEE, AICC ir IMS specifikacijomis ir jas papildo elektroninių kursų struktūrų standartizavimu, mokymo objektų daugkartiniu panaudojimu.

Mūsų atveju, norint sukurti standartizuotą e-mokymo kursą, reikia:

1. Komunikacinės sąsajos – kaip vyks bendravimas su MVS.
2. Meta duomenų aprašymo – elektroninių kurso išteklių (mokymo objektų) aprašymas, - būtinas elementas norint, kad kursą „suprastų“ MVS.
3. Komponavimas – kaip reikia aprašyti ir supakuoti elektroninį kursą, kad jis būtų visiems prieinamas.

Nors mūsų atveju kurso fiziniam realizavimui svarbiausi yra pakavimas ir meta-duomenų aprašymas, bet nereikėtų užmiršti ir komunikacinės sąsajos. Pabandydysime kiekvienu atveju parinkti mus tenkinantį standartą:

1. Komunikacinė sąsaja – labiausiai paplitusi specifikacija yra hiperteksto AICC CMI protokolas. Jis apibrėžia, kokie turi būti renkami įrašai apie studijuojantį. Kaip renkama pamokų istorija. Alternatyva šiam komunikacijos protokolui aprašyta ADL SCORM specifikacijoje. Pagrindinis skirtumas tarp AICC CMI ir ADL SCORM yra:
  - AICC CMI specifikuoja daugiau komunikacinių būdų ir elementų;
  - ADL SCORM turi griežtai apibrėžtus komunikacinius elementus.
2. Meta-duomenys – IEEE LOM – pagrindinis standartas aprašant elektroninio mokymo objektus. Remiantis šiuo standartu buvo sukurtas SCORM turinio meta-aprašymas XML formatu, t.y. aprašo praktinė realizacija.
3. Komponavimas – mokymo kursą sudaro daug įvairių mokymo objektų, kontrolinių aprašų, struktūros aprašų ir t.t., todėl reikalingas visos medžiagos surinkimas – komponavimas. Vieną iš galimų pakavimo mechanizmų aprašo IMS CP (content packing), kuris surenka visą medžiagą į vieną katalogą. Kitą būdą aprašo AICC CSF (Course structure file). Remiantis šiais abiem būdais SCORM išplėtė ir papildė savo pakavimo mechanizmą.

Apžvelgėme elektroninio mokymo standartus ir trijų mus dominančių sričių standartizavimą. Matome, kad kiekviena sritis turi savo nepriklausomus standartus, todėl atliekant mūsų užduotį kiltų papildomų suderinamumo problemų, todėl pasirenkame SCORM standartą. Nors SCORM - dar gana naujas standartas, bet pagrindinis tokio pasirinkimo privalumas tas, kad vieningame dokumente yra aprašomas visas elektroninio mokymo mechanizmas, nuo elektroninio kurso kūrimo iki atvaizdavimo.

### 1.9.5. MOKYMO TURINIO VALDYMO SISTEMŲ ANALIZĖ

Paanalizuosime keletą elektroninio mokymo sistemų, jų bendrus bruožus, daug dėmesio skirdami SCORM standarto palaikymui.

Mokymo turinio valdymo sistemos mokymo turinį pateikia atsižvelgdamos į tai, ką mokinys žino ir ko jam reikia. Tam jos, atsižvelgiant į mokinio naudojimąsi mokymo turinio valdymo sistema, sudaro mokinio žinių modelį ir, atsižvelgdamos į jį, nustato kiekvienam mokiniui tinkamiausią dalyko temų, užduočių, pavyzdžių įsisavinimo, patikrinimų seką.

Mokymo valdymo sistemos naudoja pedagogines žinias, todėl jos mokymo turinį pateikia atsižvelgdamos ne tik į mokinio žinias, bet ir į tikslus, gebėjimus mokytis vienu ar kitu būdu. Mokymo valdymo sistemos sudarytos įvairioms mokymo sritims.

Didžiausios permainos įvyko pradėjus naudoti tipines virtualias mokymosi aplinkas Learning Space, WebCT, Blackboard ir kitas. Šioms aplinkoms nebereikia kurti naujos programinės įrangos, kuriai buvo skiriama pagrindinė kurso kūrimo išteklių dalis. Jos leidžia panaudoti įvairių nuotolinių studijų būdų programinės įrangos visumą ir valdyti visą kursų organizavimą bei komunikacijos priemones. Beveik visi kursų valdymo paketai turi vartotojo sąsają, skirtą nuosekliai kursų plėtrai, tradicinių kursų elementų (plano, tvarkaraščių, paskaitų, atsiskaitymų ir t.t.) publikavimui tinkle. Dauguma jų turi dvipusės komunikacijos priemones: elektroninį pašta, elektroninę skelbimų lentą, diskusijas, pokalbius. Siūlomos ir specialios priemonės kursų valdymui: automatizuotas kurso turinio sudarymas, testų vertinimas realaus laiko veiksenoje, studentų pažangos stebėjimas ir kt.

Panagrinėsime ir išsirinksime kursų testavimui žinomesnes, pilnavertes įvairių modulių mokymo sistemas, atsižvelgsime tiek į pačios sistemos darbą, tiek į mokymo medžiagos pateikimą.

- Blackboard Learning System™ Release 6 (<http://www.blackboard.com/>) Gana paprasta sistema, kartu turinti ir pakankamai gerų įrankių, palengvinančių bendravimą nuotoliniu būdu. Pvz., konferencijai skirta virtuali klasė (Virtual Classroom), kur galima ne tik bendrauti, bet ir braižyti įvairius modelius, skirtus sistemų modeliavimui. Kita vertus sistemoje palikta gan nemažai smulkių klaidelių, kurios trukdo pilnai įsigilinti į mokymo medžiagą. Sistemoje numatoma įdiegti SCORM palaikymą.
- WebCT ([www.webct.com](http://www.webct.com)) – viena iš labiausiai paplitusių sistemų, naudojamų elektroniniam mokymui. Vienu iš rinkos lyderių WebCT tapo todėl, kad tai lanksčių, lengvai naudojamų ir pedagoginiu požiūriu patogių mokymo ir mokymosi įrankių visuma. Jau šiuo metu palaiko SCORM standartą.
- doceboLMS204 – tai atviro kodo mokymosi aplinka, viena iš pirmųjų pripažinusi ir įdiegusi SCORM standartą elektroninio mokymo sistemose. Pačios sistemos funkcionalumas mažesnis už aukščiau aprašytas, bet pakankamas vidutiniam vartotojui. Be to, trūkstamą funkciona-

lumą lengvą įdiegti. Dar vienas svarbus aspektas – nagrinėjant kodą lengviau perprasti kursų vaizdavimo ir pateikimo niuansus.

Kadangi darbe bus kuriama elektroninių kursų komponavimo sistema, palaikanti SCORM standartą, mes sukurtų kursų testavimui pasirenkame nemokamą, atviro kodo sistemą - **doceboLMS204**. Eksperimentinio kurso suderinamumui su SCORM taip pat panaudosime ir kitas atviro kodo MVS, kurias įvardinsime atliekant eksperimentą.

## 1.10. SCORM STANDARTAS

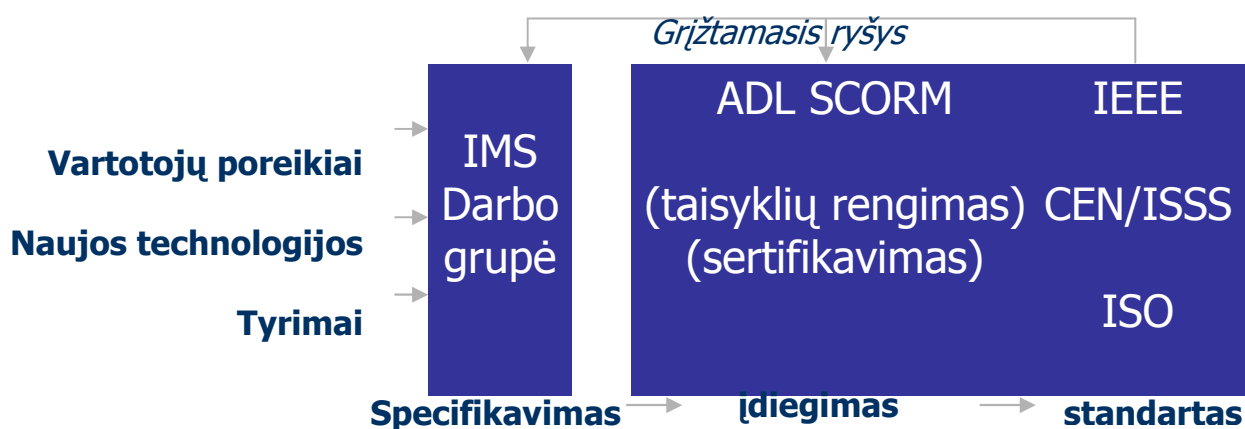
SCORM (Sharable Content Object Reference Model) – tai turinio panaudojimo objektinių modelių standartas, kuris apibrėžia Web technologijomis paremtus elektroninio mokymo objektus. SCORM leidžia padaryti mokymo objektus prieinamesnius ir naujai panaudojamus. Taip pat leidžia iš naujo panaudoti, dalintis, peržiūrėti turinį nepriklausomai nuo mokymosi aplinkos. Tokiu būdu mokymo kursas tampa nepriklausomas nuo mokymo valdymo sistemos (MVS).

Kūrėjams, kuriantiems elektroninio mokymo kursus, nebereikia išmanyti pačios pateikimo sistemos mechanizmų. Be to, esant reikalui lengva keisti aplinkas, tai sudaro ir puikias galimybes pačių kursų prekybai, nepriklausomai nuo sistemos.

Standarto naudojimo nauda:

- kurso aprašo standartizavimas;
- nepriklausomumas nuo MVS;
- padidėja elektroninio mokymo kursų kūrimo efektyvumas;
- lengvesnis daugkartinis panaudojamumas;
- kursų kūrimo kaštų mažinimas;

Standartas priimtas 2004 sausio mėnesį. 1.5 paveiksle nurodyti standartą kuriančios organizacijos ir veikiantys išoriniai veiksniai:



### 1.10.1. SCORM PAGRINDAI

Aptarsime SCORM pagrindus, būdingus iki šiol pasirodžiusioms versijoms - fundamentą.

SCORM apima šias sritis:

- *Turinio susiejimo modelis* (content aggregation model) – mokymosi objektai, paketai, meta-duomenys, manifestas, sekos ir navigavimo aprašai;
- *Vykdymo aplinka* (run-time environment) – sesijų, duomenų apsikeitimo metodai, turinio atvaizdavimas;
- *Sekos ir valdymo aprašai* (sequencing and navigation) – mokymosi valdymas, sekos ir navigavimo duomenų modeliai<sup>11</sup>.

SCORM keliami elektroniniam mokymui tikslai:

- *Prieinamumas* (accessibility) – leidžia paimti komponentus iš vienos vietos ir perkelti į kitas vietas.
- *Pritaikomumas* (adaptability) – lengvas pritaikymas prie mokymo valdymo sistemų.
- *Pajėgumas* (affordability) – galimybė padidinti efektyvumą ir produktyvumą.
- *Išliekamumas* (aurability) – paprastas sistemos atnaujinimas tobulėjant standartui.
- *Sąveikingumas* (interoperability) – galimybė panaudoti vieną komponentą keletą kartų, iš bet kurios vietos, nepriklausomai nuo platformos ir įrankių.
- *Panaudojamumas* (reusability) – galimybė naudoti tą patį komponentą daug kartų.

Pagrindinės sąvokos, naudojamos SCORM standarte:

- *Mokymo objektas* (assets)<sup>12</sup> – tai media failai, naudojami kuriant mokymo kursus, pvz. mp3, jpg ir t.t.
- *Mokymo turinio objektas* (sharable content object) – tai mokymo objektų rinkinys, sudarytas iš mokymo objektų, kuris gali būti pateiktas kaip kurso dalis, jis parodo kaip elgtis MVS.
- *Meta duomenų komponentas* – apibrėžia duomenis, jų paskirtį.
- *Turinio pakavimas* (content packing) - surinkus reikiamus turinio komponentus, juos reikia apjungti ir organizuoti. Tam naudojamas turinio pakavimas. Paketas gali apimti visą kursą ar jo dalį.

Plačiau panagrinėsime, kaip SCORM sprendžiama turinio pakavimo problema ir svarbiausią elementą – manifesto aprašą.

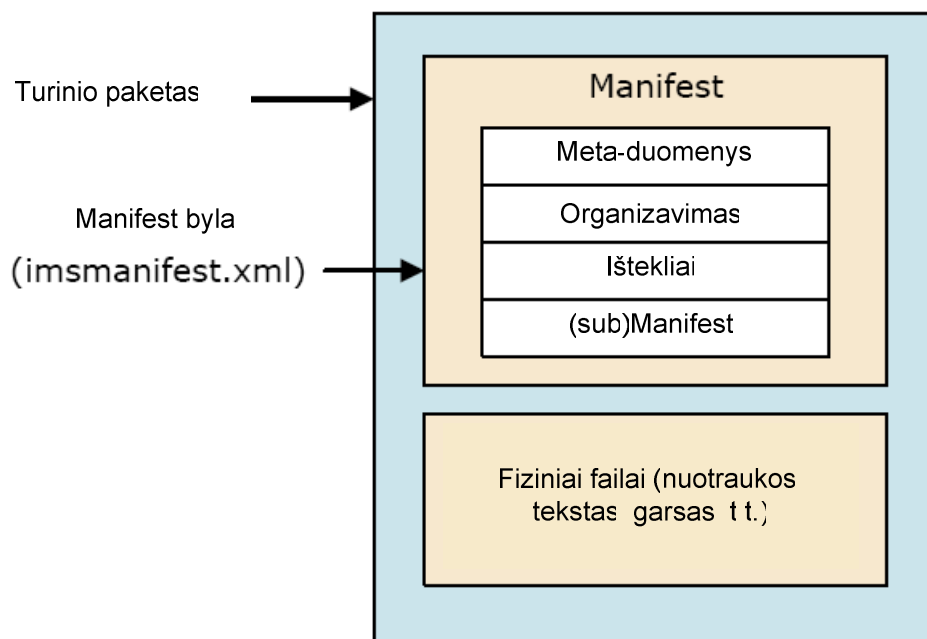
*Fizinį paketo failą* (package interchange file) sudaro visi valdymo failai, manifestas ir mokymo turinio failai, sudaryti iš mokymo objektų. Failas suglaudinas ZIP formatu.

Turinio pakavimas sudarytas iš dviejų pagrindinių dalių (6 paveiksle pateikta paketo struktūra):

- XML dokumento (*manifest file*), aprašančio visą turinį, jo struktūras, apjungimą, sąsajas, vadinamu manifestu.
- Fizinį failų, kuriuose saugomas visas kursų turinys<sup>13</sup>.

<sup>11</sup> SCORM® 2004 2nd Edition Overview. ADL Technology Center, 2004. P.28.

<sup>12</sup> SCORM® Version 1.3.1 Content Aggregation Model. ADL Technology Center, 2004. P.21.



1.6 pav. Turinio paketo struktūra

Manifestas – XML dokumentas, kuriame yra aprašytas visas paketo turinys ir nurodyta, kaip visas paketas organizuotas ir struktūrizuotas. Kiekvienas paketas turi pagrindinį manifestą ir gali turėti vieną ar daugiau sub-manifestų. Manifesto failas pakete būtinai pavadinamas *imsmanifest.xml* ir talpinamas pagrindinėje paketo direktorijoje. Fizinė *imsmanifest.xml* realizacija pateikta 1.7 paveiksle.

## imsmanifest.xml

```

<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" identifier="MANIFEST-65A1BA8F-3A73-479E-0F63-163FFDA5B0E1"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
  http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd">
  <metadata />
  <organizations default="ORG-9CD8FA62-FF8C-B998-55CA-C1599862AEA6">
    <organization identifier="ORG-9CD8FA62-FF8C-B998-55CA-C1599862AEA6" structure="hierarchical">
      <title>Organization</title>
      <item identifier="ITEM-CD55CF07-609F-A592-E0B9-8EC68A32942F" invisible="true" identifierref="RES-5669B8B0-85B5-01D8-0A33-3284AE5A6218">
        <title>sundialjelsim</title>
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="RES-5669B8B0-85B5-01D8-0A33-3284AE5A6218" type="webcontent" href="sundialjelsim.html">
      <file href="images/jspowered.gif" />
      <file href="images/solarGeom/sun.gif" />
      <file href="jscore.jar" />
      <file href="progress.jar" />
      <file href="SolarGeom.jar" />
      <file href="sundialjelsim.jsm" />
      <file href="sundialjelsim.html" />
    </resource>
  </resources>
</manifest>

```

*(kaip interpretuoti paketa)*

*(metaduomenis)*

*(kaip organizuojamas turinys)*

*(reikalingi resursai)*

1.7 pav. Manifesto failo struktūros pavyzdys

### 1.11. ELEKTRONINIO MOKYMO SISTEMOS FUNKCINIS MODELIS

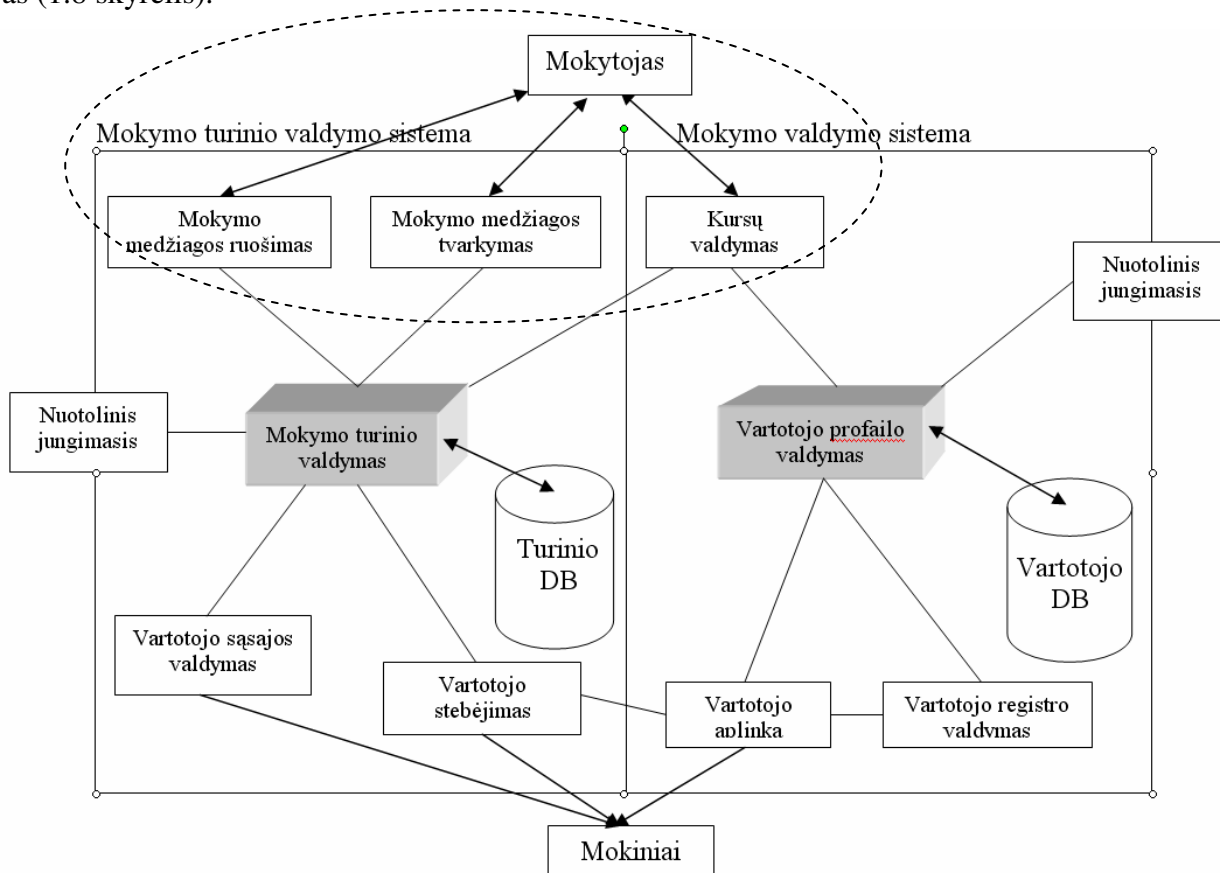
Elektroninio mokymo sistemą galima suskaidyti į dvi subsistemas (1.8 paveikslas):

- Mokymo turinio valdymo sistema;
- Mokymo proceso valdymo sistema.

Darbe plačiau nagrinėjama sritis 1.8 paveiksle pažymėta punktyrine linija:

- Mokymo medžiagos ruošimas;
- Mokymo medžiagos tvarkymas;
- Iš dalies – kursų valdymas.

Visus šiuos procesus turėtų kontroliuoti mokytojas (pedagogas) – mokomosios medžiagos rengėjas (1.8 skyrelis).

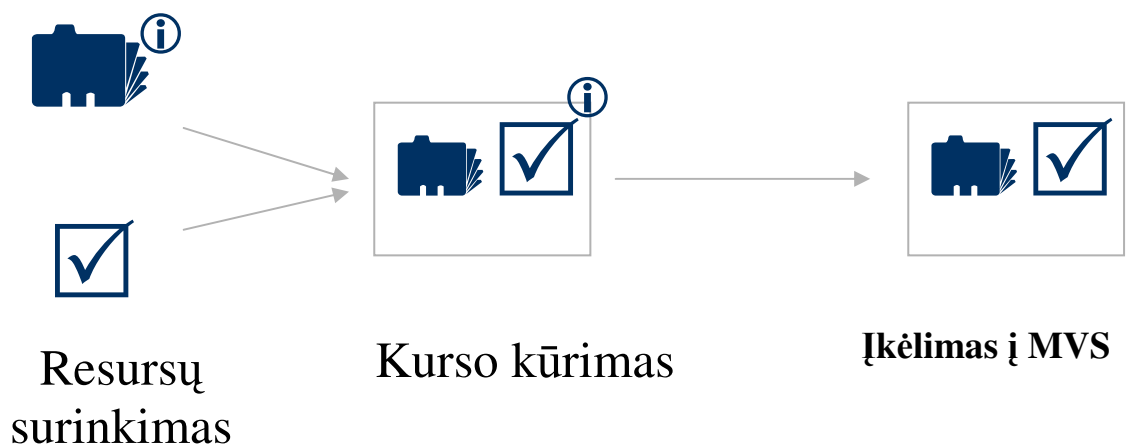


1.8 pav. Elektroninio mokymo bendra funkcinė schema

### 1.12. ELEKTRONINIŲ KURSŲ MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMOS SPECIFIKAVIMAS

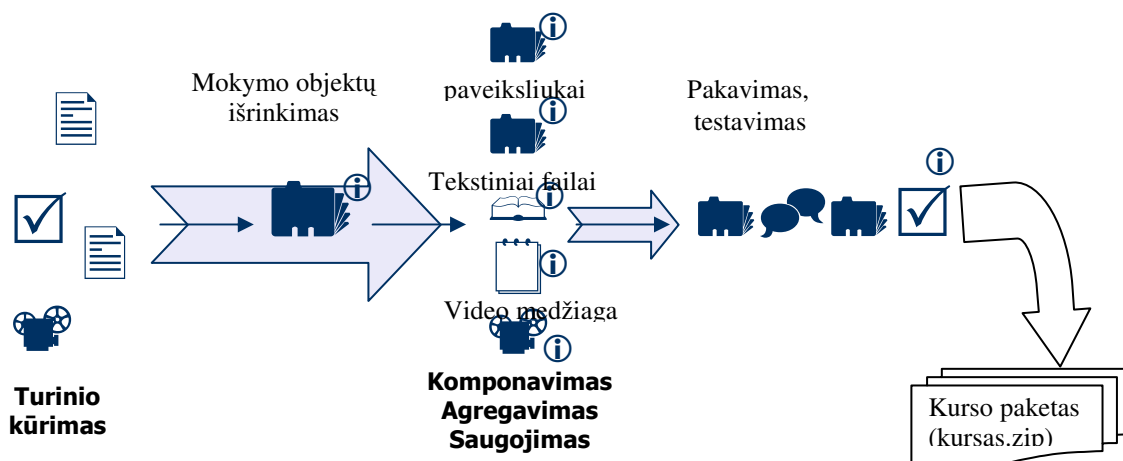
Žemiau pateiktame paveiksle (1.9 pav.) pavaizduotas kurso kūrimo pateikimo procesas. Pirmiausia kuriant kažkokį mokymo kursą surenkame medžiagą, po to ją apdorojame ir pateikiame galutiniam vartotojui – mokiniai.





1.9 pav. Elektroninio kurso atsiradimo kelias

Sekančiame paveiksle (1.10 pav.) parodyti detalesni procesai, vykstantys **kurso komponavimo** etape. Kursų komponavimo sistemos veikla prasideda iškart po turinio, mokymo objektų kūrimo. Sistemos rezultatas – SCORM standartu aprašytas ir sukomponuotas elektroninio mokymo kursas.

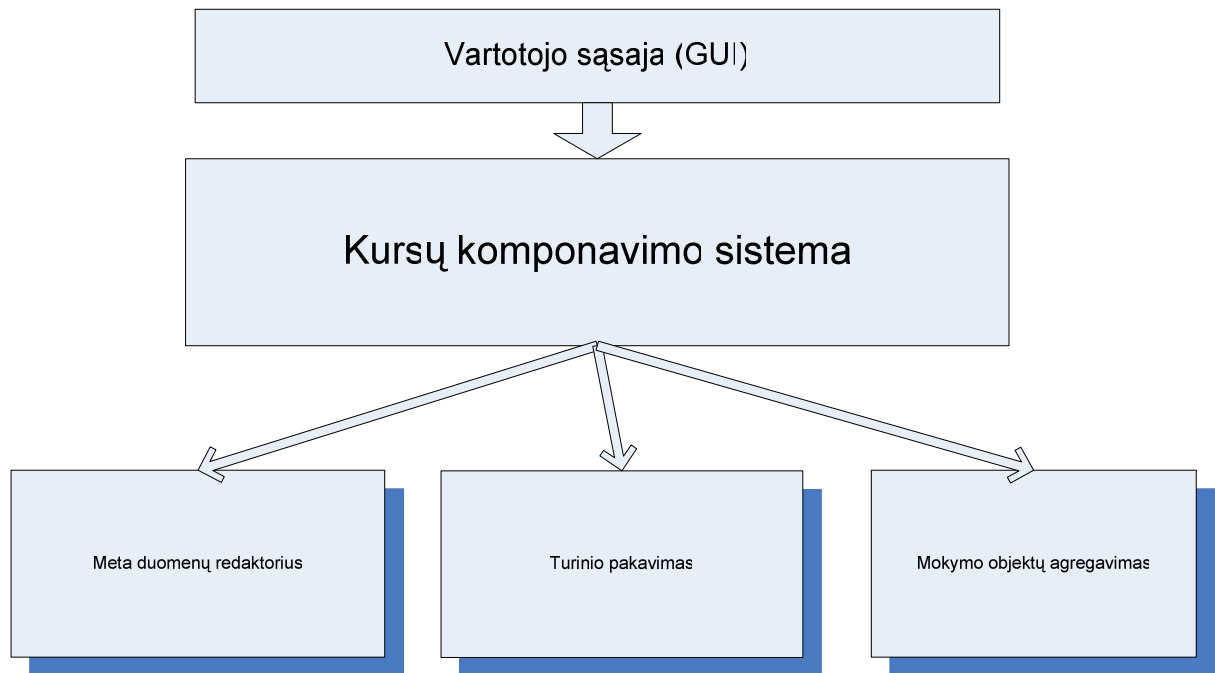


1.10 pav. Elektroninio kurso kūrimo ir komponavimo schema

Žemiau pateiktame paveiksle (1.11 pav.) pavaizduota numatoma mokymo objektų komponavimo sistemos architektūra.

Elektroninių kursų mokymo objektų komponavimo sistemą sudaro:

- Vartotojo sąsaja;
- Kurso kūrimo terpė:
  - Meta duomenys – mokymo objektų aprašai;
  - Turinio pakavimas – galutinis turinio paruošimas publikavimui;
  - Mokymo objektų agregavimas – kaip mokymo objektai bus organizuojami ir kokia tvarka pateikiami vartotojui.



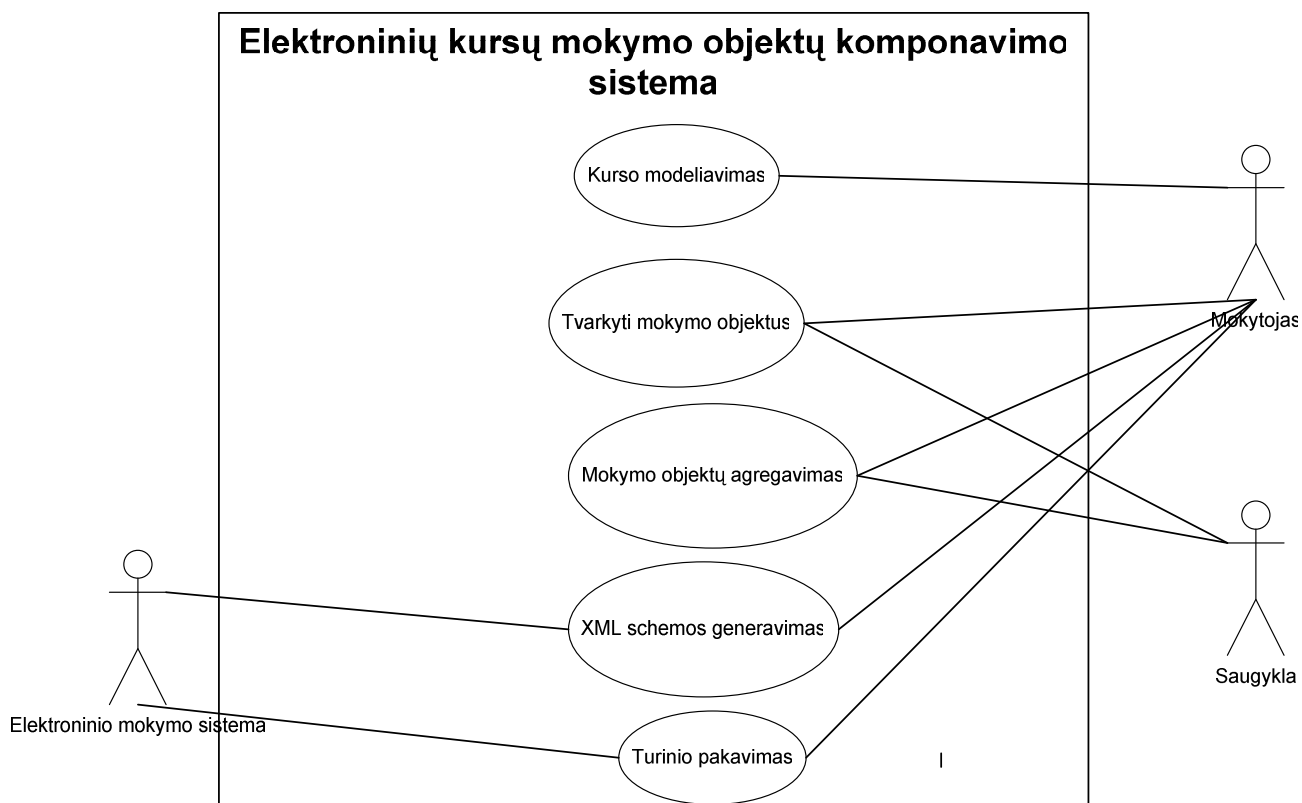
**1.11 pav.** Kursų mokymo objektų komponavimo sistemos architektūra

Formaliai galime apibrėžti tokio įrankio teikiamą naudą:

- nereikia tiesiogiai dirbti su XML failais;
- greitesnis XML struktūros kūrimas;
- lengvas kursų redagavimas;
- standarto plėtojimas;
- platus suderinamumas.

### 1.13. KOMPIUTERIZUOJAMOS SISTEMOS FUNKCIJOS

Pabandysime aiškiai apibrėžti kompiuterizuojamos sistemos funkcijas. Žemiau esančiame paveiksle (1.12 pav.) pateikiamas numatomas elektroninio mokymo sistemos panaudojimo atvejų modelis.



1.12 pav. Elektroninių kursų mokymo objektų komponavimo sistemos panaudojimo atvejai

1.1 lentelė – Panaudojimo atvejų paaiškinimas

Panaudojimo atvejis	Detalizavimas
Kurso modeliavimas	Pradiniai kuriamo kurso nustatymai, duomenys, apimtis, vardai ir pan.
Tvarkyti mokymo objektus	Mokymo objektų tvarkymas, išrinkimas, kūrimas
Kurso agregavimas	Nustatome kurso struktūrą
Kurso komponavimas	Kurso aprašų sukūrimas ir surinkimas, pakavimas
XML schemas generavimas	Generuojamas manifesto failas, pagal sukurtą kurso struktūrą ir duomenis
Kurso pateikimas	Supakuoja kursą standarto SCORM nurodytu formatu ir pateikia galutiniam vartojimui

#### 1.14. FUNKCINIAI IR NEFUNKCINIAI REIKALAVIMAI, APRIBOJIMAI

Elektroninio mokymo sistemos pagrindinė paskirtis – pagelbėti įvairaus amžiaus žmonėms įsisavinti nuotoliniu būdu pageidaujama temų kursą, įgyti naujų teorinių žinių, formuoti ar įtvirtinti praktinius įgūdžius, o taip pat patikrinti turimus ar įgytus gebėjimus, realizuojant grįžtamojo ryšio principus.

Labiausiai tikėtinas sistemos vartotojas – neturintis profesionalių žinių informacinių technologijų srityje, bet profesionalas pedagogikoje. Atsižvelgiant į tai, sudaromi nefunkciniai reikalavimai sistemai:

- Paprastumas – darbas su elektroninio mokymo kursų kūrimo sistema turi būti nesudėtingas, nereikalaujantis iš vartotojo gilių žinių informacinių technologijų srityje. Konkrečiu mūsų atveju vartotojas dirba su grafine vartotojo sąsaja, o ne su XML failais ir mokymo objektais tiesiogiai.
- Patikimumas – sistema turi būti patikima, t.y. korektiškai atlikti jai pavestas užduotis ir funkcijas.
- Efektyvumas – sistema turi atlikti jos vykdomas funkcijas greičiau, negu tai būtų atliekama be sistemos, ir kad jos atliekamos funkcijos ir jų atlikimo sparta būtų realiai naudingi sistemos vartotojams.
- Nesudėtingas eksploatavimas – sistemos naudojimas neturi reikalauti didesnių papildomų žmoniškųjų išteklių bei materialinių resursų.
- Saugumas – vartotojai, naudodami šią sistemą, turi jaustis saugiai dėl savo įvestų duomenų.

#### ***1.14.1. REIKALAVIMAI STANDARTAMS***

Pagrindinis standartas, kurio bus griežtai prisilaikoma, – SCORM 1.2. Nors jau pasirodė ir naujesnė standarto versija – SCORM 2004, senesnę versiją naudosime dėl paprastesnio realizavimo ir platesnio paplitimo mokymo valdymo sistemose, - tai bus itin svarbu atliekant eksperimentą. Taip pat atsižvelgsime ir į informacinių technologijų mokymo standartus, stengiantis pateikti būtent tai, ko reikia.

Sukurti kursai turi būti pilnai suderinti su SCORM 1.2 standartą palaikančiomis mokymo valdymo sistemomis (pvz., WebCT, dokebosLMS).

#### ***1.14.2. REIKALAVIMAI VARTOTOJO SĄSAJAI***

Vartotojo sąsaja turi būti paprasta, lengvai perprantama, intuityvi. Vartotojui informacija turi būti pateikiama aiškiai ir suprantamai. Galimos klaidos apdorotos vengiant paprastam vartotojui nesuprantamų pranešimų. Navigavimo sistema logiškai pagrįsta ir intuityvi. Sistemos rezultatai pateikiami aiškiai ir suprantamai.

#### ***1.14.3. REIKALAVIMAI DUOMENIMS***

Programai naudojamus duomenys galima suskirstyti į tokias dalis:

- Mokymui naudojami objektai – tai įvairūs paveikslėliai, schemas, garso failai ir pan.
- Programos generuojami duomenys – tai gaunami programos rezultatai, XML failai, aprašantys duomenų struktūras, ir galutinis kurso paketas, kuris glaudinamas ZIP formatu (PIF).

### 1.15. KŪRIMO ĮRANKIŲ PASIRINKIMAS.

Projektavimui pasirenkame MS Visio paketą, nes mums jis lengviausiai prieinamas, be to, nereikės aiškintis įrankio veikimo principų.

Darbo praktiniam realizavimui, apsvarsčius visas galimybes, atsižvelgiant į praktinį realizuotos sistemos naudingumą bei turimas galimybes, apsistota ties šiais galimais variantais:

1. JAVA programavimo kalba. Galimos pasirinkimo priežastys:
  - būsimos sistemos nepriklausomumas nuo platformos;
  - objektinis programavimas;
  - galimybės, leidžiančios lengvai pritaikyti būsimą sistemą web aplinkai;
  - šio darbo autoriaus praktiniai įgūdžiai dirbant su šia programavimo kalba;
  - daug įvairių derinimo įrankių.
2. OpenScrip programavimo kalba. Galimos pasirinkimo priežastys:
  - OpenScript kalba panaši į Java, todėl programuojant sunkumų neturėtų kilti.
  - ToolBook Instructor palaikantis OpenScript yra specializuotas įrankis, skirtas elektroninių kursų kūrimui ir palaikantis XML standartą bei kai kuriuos elektroninio mokymo standartus, todėl mes galime be jokių papildomų priemonių generuoti XML struktūras.
  - ToolBook Instruktor leidžia patogiai ir lanksčiai kurti vartotojo sąsają.
  - Funkcijų, skirtų SCORM standartui panaudojimo galimybė.

Pasirenkame OpenScrip programavimo kalbą ir ToolBook Instructor vartotojo sąsajos kūrimo įrankį dėl jau anksčiau minėtų priežasčių. Nors ToolBook Instructor - komercinis produktas, jis reikalingas tik įrankio kūrimo stadijoje; programos naudojimui reikalingas įskiepis Neuron yra nemokamai prieinamas ir tinka visoms Windows sistemoms. Taip pat bus praplėstos žinios susipažįstant su naujais kūrimo įrankiais, specializuotais elektroniniam mokymui.

Sistemos sukurtų mokymo paketų testavimui naudosime atviro kodo elektroninio mokymo sistemą dokebosLMS2004, (šio pasirinkimo priežastys aptartos 1.9.5 skyrelyje).

### 1.16. DARBO TIKSLAS IR SIEKIAMŲ PRIVALUMŲ

Šio darbo tikslas - sukurti pilnavertę elektroninių mokymo objektų komponavimo sistemą. Sistemos funkcionalumui bei naudingumui įrodyti ir testavimui atlikti sukursime elektroninį mokymo kursą, atitinkantį XI-XII klasių Interneto temas.

1.2 lentelė – Sistemos privalumai

Komponavimo sistemos privalumai	Lyginant su kitomis panašiomis sistemomis
Objektinis kurso kūrimas	Nėra griežtai apibrėžti mokymo objektai
Lengvai perprantama, paprasta vartotojo sąsaja	Daug funkcijų, daug problemų
Standarto SCORM palaikymas	Suderinamumo trūkumas
Nepriklausomumas nuo MVS	Integracija į MVS

Realiai bus atlikti šie darbai:

- Suprojektuota Interneto temų elektroninių kursų mokymo objektų komponavimo sistema, palaikanti SCORM standartą (projektas);
- Realizuotas elektroninių kursų mokymo objektų komponavimo sistemos vieno mokymo objekto aprašymas ir paketo sukūrimas SCORM standartu (programa);
- Sukurtas Interneto temų e-mokymo demonstracinis kursas (pilnas kurso paketas aprašytas SCORM 1.2 standartu ir pateiktas PIF failu);
- Realūs ir išsamūs sistemos bandymai, naudojant kursų pateikimui docebOLMS204 MVS (testavimo rezultatus).
- Sukurto kurso suderinamumo eksperimentas naudojant kuo daugiau įvairių MVS, kurios palaiko SCORM 1.2 standartą.

### 1.17. ANALIZĖS IŠVADOS

Atlikus pirminę analizę t.y.:

- apibendrinus elektroninio mokymo teoriją;
- apžvelgus šiuo metu egzistuojančias elektroninio mokymo sistemas;
- apžvelgus elektroninio mokymo standartus;

Galime teigti, kad mūsų elektroninių mokymo objektų komponavimo sistema:

- leis lengviau aprašyti kursus XML kalba pagal SCORM standartą;
- suteiks galimybes prieiti prie viešų mokymo objektų saugyklų;
- sistema ir kursas taps nepriklausomi nuo elektroninės mokymo sistemos;
- tai leis pateikti pilnavertį elektroninį mokymo kursą kaip prekę.

Papildomai galime pridėti, jog darbo tikslas - ne tik sukurti elektroninių kursų mokymo objektų komponavimo sistemą, pateikti interneto temų SCORM 1.2 aprašytą kursą, bet ir pagilinti žinias informacinių sistemų kūrimo ir elektroninio mokymo srityse.

## 2. ELEKTRONINIŲ KURSŲ MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMOS PROJEKTAVIMAS

Universalios sistemos, galinčios kurti viską – teikti kursus, juos kurti, modeliuoti mokymo aplinką, atlikti testavimą ir kitas elektroninio mokymo funkcijas leidžia operatyviai, objektyviai ir kokybiškai pateikti mokymo kursus ir kitas su tuo susijusias paslaugas. Bet dėl to prarandamas lankstumas, leidžiantis pasirinkti reikiamus ir naudingus sistemos komponentus, greitai diegti naujoves. Todėl masyvos, viską apimančios elektroninio mokymo sistemos turės būti surenkamos iš atskirų komponentų - modulių. Pirmas žingsnis jau padarytas – priėmus SCORM standartą ir jį įdigus, kursai tampa nepriklausomi nuo elektroninės mokymo sistemos.

Remdamiesi SCORM standartu, suprojektuosime ir realizuosime elektroninio mokymo kursų mokymo objektų komponavimo sistemą. Sistemos pagalba bus galima aprašyti elektroninį kursą ir pateikti išeities failą PIF, kurį galima patalpinti ir publikuoti bet kurioje MVS, palaikančioje SCORM 1.2 standartą. Projektuodami remsimės analizės gauta medžiaga ir aprašytais reikalavimais.

Pirmiausia tiksliai apibrėšime pagrindinius sistemai keliamus reikalavimus, kuriais vėliau remdamiesi projektuosime būsimą sistemą. Pateiksime realizacijos schemą ir vartotojo sąsajos langų prototipus.

Projekto tikslas - suprojektuoti ir realizuoti elektroninių kursų mokymo objektų komponavimo sistemos modulį pagal išanalizuotus ir aprašytus principus. Su sukurta realizacija atlikti eksperimentą, kurį detalizuosime ir smulkiau aptarsime eksperimentinėje dalyje.

Projektavimui pasirinktas MS Viso 2003, t.y. priemonė, kurios mums pilnai turėtų pakakti braižant UML schemas. Pasirinkimo priežastys aptartos analizės dalyje.

Realizaciją, kaip anksčiau aptarėme, atliksime OpenScript programavimo priemonėmis.

### 2.1. KOMPONAVIMO PROCESO DETALIZAVIMAS

Konkretizuosime anksčiau jau nagrinėtus komponavimo procesus (1.4 pav.). Patikslinsime **aplinkos analizėje** vaizduojamus **proceso** modelius iki smulkiausių procesų. Veiksams apibrėžti sudaromas panaudojimo atvejų modelis (2.1 pav.). Paaiškinimai pateikti 2.1 lentelėje.

2.1 lentelė - Uždavinių-procesų aprašymas

Uždavinys	Aprašymas
Kurti kurso šabloną	Vartotojui pateikiami struktūrizuoti mokymo objekto, elektroninio kurso atvaizdavimo šablonai. Realizacijoje pateiksime standartinį mokymo objekto atvaizdavimą html faile, kurį aptarsime vėlesniuose skyreliuose.
Aprašyti kursą	Vartotojas aprašo mokymo objektus, kursą metaduomenimis, pa-

	gal kuriuos yra vykdoma interpretacija.
Išrinkti mokymo objektus	Vartotojas nurodo norimą mokymo objektą, greitesniam išrinkimui naudojami filtrai, atrenkantys norimo tipo mokymo objektus.
Surinkti bylas	Visos bylos, reikalingos mokymo objekto komponavimui atlikti, surenkamos į vieną vietą – katalogą.
Pradinių duomenų valdymas	Pradinių nuostatų keitimas, parinkimas.
Mokymo objektų agregavimas	Išrinktų mokymo objektų organizacinės struktūros kūrimas, eiliškumo, pasirodymo trukmės nustatymas.
XML schemų generavimas	Generuojamas mokymo objektų aprašo failas – manifest.xml ir metaduomenų aprašo failas metadata.xml. SCORM standartas numato, jog metaduomenys gali būti ir manifesto faile.
Turinio pakavimas	Visi reikalingi kurso ar mokymo objektui pateikimui failai surenkami ir suarchyvuojami kaip PIF failas ZIP formato PKZip v2.04g archyvas, atitinkantis RFC 1951 reikalavimus.
ZIP archyvavimas	Leidžia suarchyvuoti turinio paketą į PIF failą.



2.1 pav. Komponavimo proceso detali panaudojimo atvejų schema



Kaip matome iš schemos (2.1 pav.), pagrindinė sistemos funkcija – vartotojo (Mokytojo) nurodytą mokymo objektą ar objektus sukombonuoti tarpusavyje (agregavimas, išrinkimas) ir SCORM standarto nurodytais aprašais (manifesto failu) pateikti kurso PIF failą.

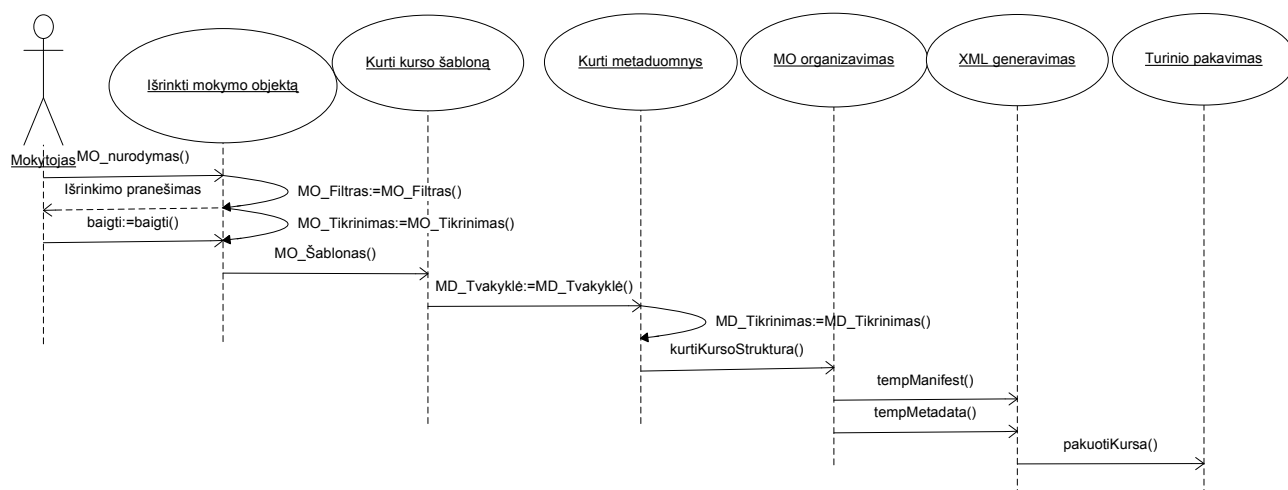
## 2.2. KOMPONAVIMO SISTEMOS ELGSENOS APRAŠYMAS

Kaip jau minėjome, komponavimo sistemai vartotojas nurodo mokymo objektus, parenka tam tikrus nustatymus ir rezultatu gauną paketą, aprašytą SCORM standartu, kuris pateikiamas viename nurodytame kataloge ar PIF failu (ZIP formatas).

Aprašysime, ką sistema turėtų atlikti, norėdama sėkmingai sukurti mokymo paketą:

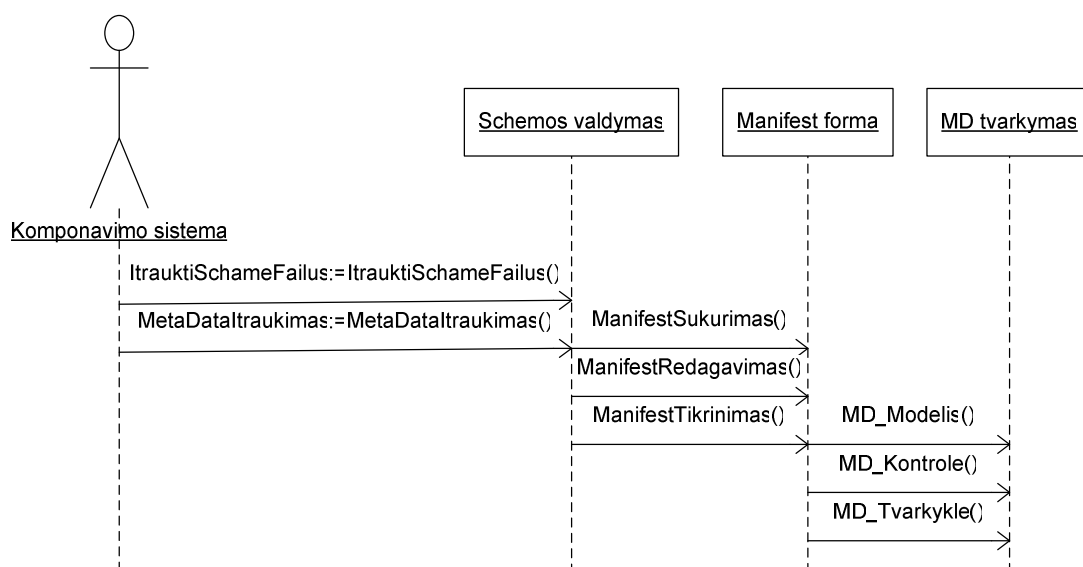
1. Leisti nurodyti mokymo objektą ar objektų grupę, iš kurių kuriamas komponentas. Išrinkimo palengvinimui naudojami filtrai.
2. Įtraukti į paketą visus failus, kurie tiesiogiai yra susiję su mokymo objektu. Pvz., jei mokymo objektas yra html byla, programa turėtų surinkti ir toje byloje naudojamus objektus.
3. Įkelti į paketą visus schemos failus, reikalingus XML aprašo palaikymui.
4. Sugeneruoti mokymo atvaizdavimo šabloną MVS.
5. Leisti redaguoti metaduomenis (konkrečiau metaduomenis aptarsime specifikuodami SCORM reikalavimus).
6. Leisti kurti mokymo objektų organizacines struktūras, schemas.
7. Paruošti paketo vietą – katalogą. Papildomai sistema šioje stadijoje turėtų atlikti šiuos veiksmus:
  - a. Tikrinti, ar nurodytoje vietoje yra reikalingas kiekis atminties paketo saugojimui;
  - b. Tikrinti, ar paketo kataloge yra kitų failų; jei taip, turėtų juos pašalinti, nes kitu atveju jie gali būti neaprašyti manifesto faile;
  - c. Tikrinti, ar toje vietoje pakankamos teisės atlikti reikalingas operacijas.
8. Atlikti sukurtų mokymo aprašų perkėlimą į XML formatą – XML generavimas.
9. Turinio pakavimas – surinkti darbo rezultatus į vientisą paketą. Šiame etape turėtų būti galima:
  - a. Leisti peržiūrėti sukurto manifesto failą;
  - b. Sukurti paketo PIF failą (ZIP glaudinimas).

Norėdami tiksliau identifikuoti ir apibrėžti sistemos veiklos operacijas, paanalizuosime komponentų sąveikas. 2.2 paveiksle pateikiamas schematinis vartotojų veiksmų modelis, kai buvo pasirinktas elektroninių kursų mokymo objektų komponavimas.



2.2 pav. Mokymo objekto komponavimo sekos diagrama

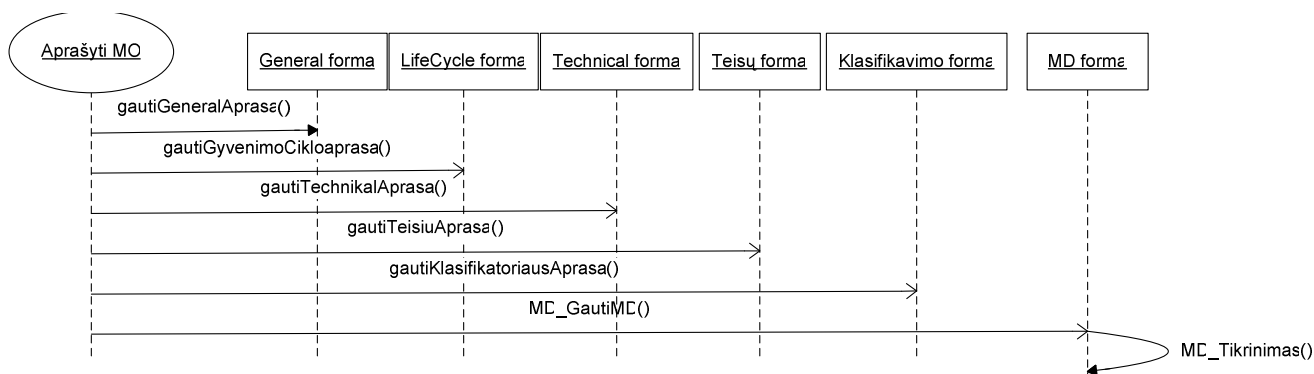
Papildydami pagrindinę komponavimo sekos diagramą, tikslesniam veiklos identifikavimui papildomai panagrinėsime metaduomenų aprašymo etapą.



2.3 pav. Metaduomenų ikėlimas

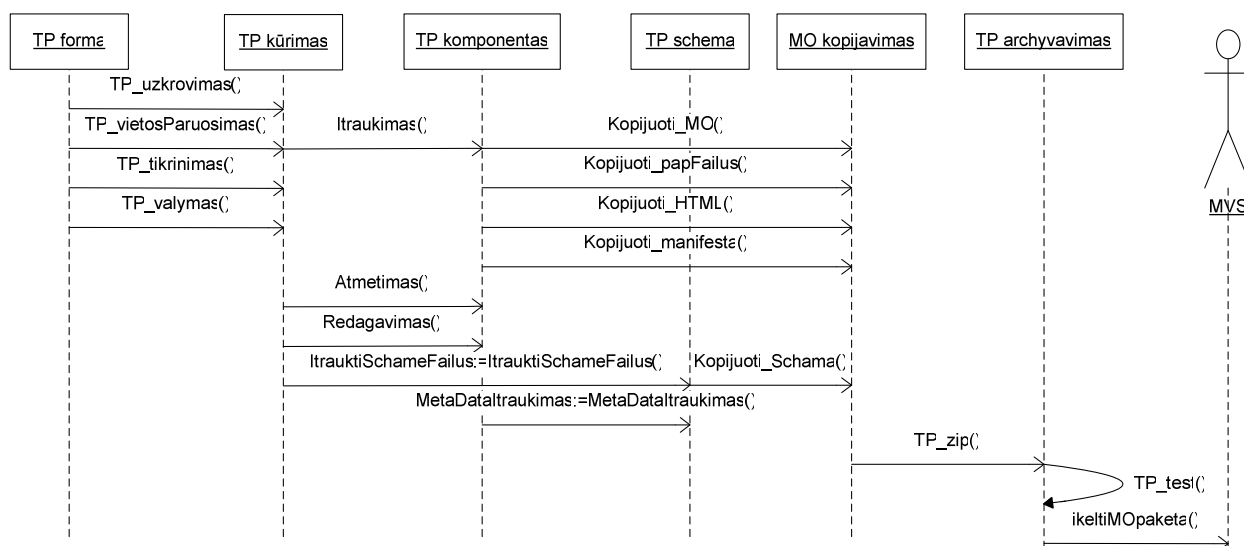
2.4 paveiksle parodyta, kokios formos turėtų būti užpildytos. Realioje situacijoje pildymo tvarka nesvarbi. Užpildžius formas, sistema turėtų surinkti gautus duomenis, patikrinti, ar duomenys suderinti su SCORM standarto reikalavimais, ir apie tai informuoti vartotoją.

Aprašydami metaduomenis, schemeje identifikuojame tik tas formas, kurios turėtų būti aprašant mokymo objektus. Smulkiau kiekvieną formą panagrinėsime ir specifikuosime išskirdami SCORM standarto reikalavimus.



2.4 pav. Metaduomenų aprašymas sekų diagrama

Taip pat papildomai panagrinėsime ir turinio pakavimo etapą (2.5 pav.). Šiame etape yra surenkama visa reikalinga informacija, atliekami standartiniai veiksmai, būtini turinio paketo paruošimui.



2.5 pav. Turinio pakavimo sekų diagrama

### 2.3. SCORM STANDARTO REIKALAVIŲ IŠSKYRIMAS

Šiame skyrelyje pabandydysime išskirti mūsų darbui svarbius SCORM standarto reikalavimus. Tai yra tik tas specifikacijas, kurios yra reikalingos norint pilnai aprašyti ir sukurti elektroninio mokymo kursą, suderintą su šiuo standartu. Didelį dėmesį skirsime IMS pakavimo ir metaduomenų specifikacijai. Realizavimui naudosime SCORM 1.2 versiją, nes ji šiek tiek paprastesnė už naujesniąją SCORM 2004, be to, ši versija yra stabili ir daugiau išbandyta, plačiau paplitusi. Atliekant testavimą tai bus itin svarbu, nes daugelis MVS kol kas palaiko tik 1.2 versiją.

Su SCORM standartu suderintas elektroninis mokymo turinys - tai:

- aprašytas metaduomenimis;
- prieinamas per Web naršyklę;
- struktūrizuoti aprašyti mokymo objektai (SCO);
- lengvai perkeliamas iš vienos MVS į kitą;
- išeities failas PIF, atitinkantis RFC 1951 reikalavimus.

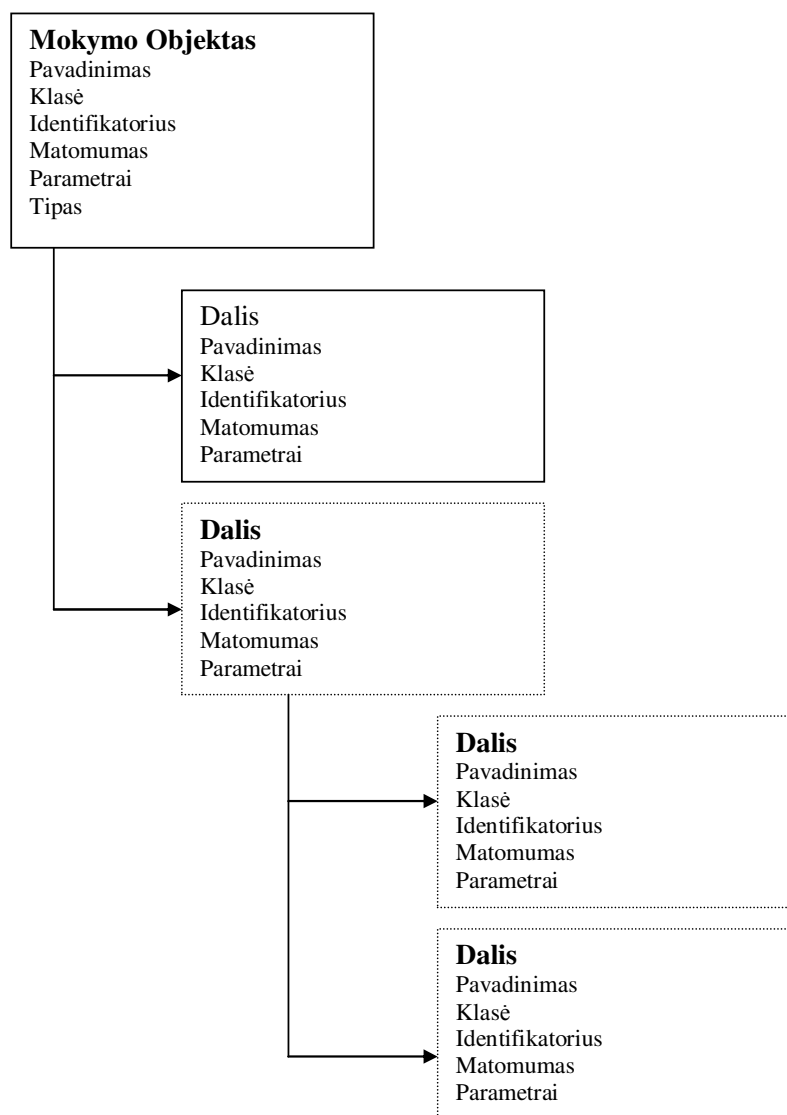
SCORM suderinamumas su mokymo objektais:

- Teoriškai gali būti aprašyti visi failai, bet pagrindą sudaro HTML bylos. Taip pat plačiai paplitę ir Word ar Acrobat pdf failai kaip mokymo objektai.
- Galima išskirti elektroninio mokymo duomenų objektus, kurie yra naudojami kituose mokymo objektuose kaip pagalbiniai, pvz., paveikslėliai, schemas.

Mokymo objektai dar gali būti skirstomi į „online“ ir „offline“. Darbe naudosime tik „offline“ mokymo objektus, todėl plačiau šios srities nenagrinėsime.

### 2.3.1. MOKYMO OBJEKTŲ ORGANIZAVIMAS IR ATVAIZDAVIMAS

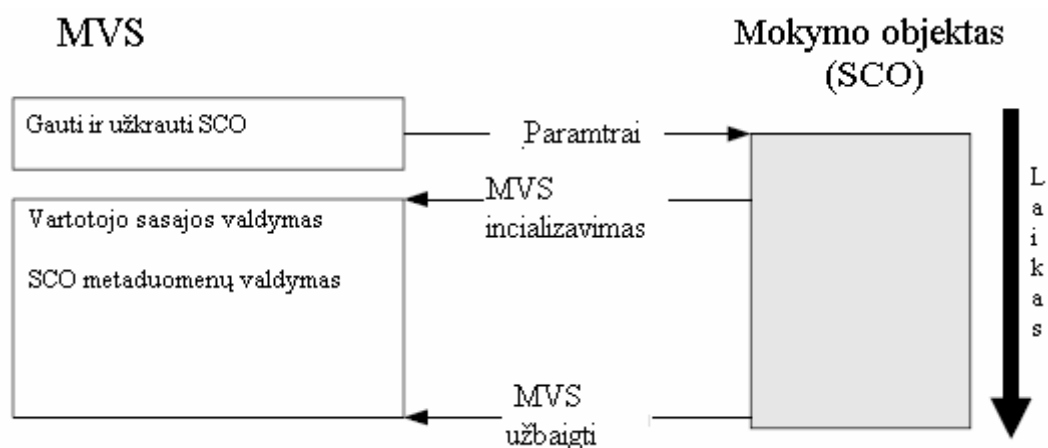
Mokymo objektų organizavimas. SCORM 1.2 versija nurodo, kokia tvarka mokymo objektai bus naudojami, todėl vartotojas turi pats tai nuspręsti, o sistema turi leisti keisti eiliškumą. Reikėtų pasitebėti, kad tiems patiems mokymo objektams gali būti sukurtos kelios organizavimo schemas. Naujesnėje SCORM 2004 versijoje ši sritis yra labiau išplėtotą.



2.6 pav. Elektroninių kursų mokymo objektų organizavimas

Mokymo objektai yra struktūrizuojami hierarchinėmis medžio struktūromis. 2.6 pav. pavaizduota smulkesnė mokymo organizavimo struktūra, kurioje dar parodoma ir kas aprašo mokymo objekto elgesį. Sistemoje mes naudosime vieną standartinį mokymo objektų organizavimo šabloną, atitinkantį 2.6 paveikslą tik hierarchija apribosime iki 2 lygių.

Norint, kad elektroninis kursas būtų gerai atvaizduotas, MVS apibrėšime kaip yra pateikiamas kursas. Mokymo objekto atvaizdavimas MVS parodytas paveiksle 2.7. Jei mokymo objektas turėtų savyje papildomų objektų, tai jie būtų įkeliami po MVS iniciavimo atliekant tuos pačius žingsnius.



2.7 pav. Mokymo objekto atvaizdavimas MVS

Smulki elektroninio mokymo kurso, aprašyto SCORM standartu, sąveika tarp mokymo valdymo sistemos pateikiama 8.4 priede. Schemose nurodomas kurso atvaizdavimas tarp kliento ir serverio, remiantis Ostyn C.

SCORM standartu aprašytą mokymo objektą atvaizduosime pasinaudodami HTML puslapiu su kadrais (frames) (pav. 17). Toks atvaizdavimas reikalingas norint testuoti įvairius mokymo objektus, o ne tik jau aprašytus kursus. Pvz., norint testuoti pasirinktoje MVS grafinę schemą ar paveikslėlį. Kitas šio sprendimo privalumas tas, kad mes galime patikrinti sistemos duodamus rezultatus (paketo tinkamumą, dalinį funkcionalumą), apsieidami ir be MVS. Šio sprendimo realizavimui pasinaudosime Ostyn C. pateiktu techniniu sprendimu<sup>14</sup>.



2.8 pav. Mokymo objekto atvaizdavimas naršyklėje

<sup>14</sup> Ostyn C., Cooking up a SCORM: A SCORM 1.2 Content Cookbook for Developers. 2001, Click2Learn. P.55.

### 2.3.2. *METADUOMENŲ APRAŠYMAS IR MANIFESTO FAILO GENERAVIMAS*

Norint sugeneruoti XML manifesto failą reikia:

- Pridėti metaduomenis, aprašančius SCORM palaikymą;
- Pridėti resursų aprašymą;
- Pridėti organizavimo aprašymą;
- Pridėti metaduomenis, aprašančius SCORM 1.2 techninius paketo aprašymus;
- Pridėti metaduomenis, jei yra mokymo objektų tarpusavio priklausomybė.

Metaduomenų aprašymą išskirstysime į grupes, nes taip lengviau bus aprašyti ir realizuoti, generuoti XML, kurti vartotojo sąsają. XML aprašai turi atitikti SCORM schemą, valdymo failų struktūras. Naudotų metaduomenų aprašymo XML aprašų paaiškinimai ir svarbesnės grupės pateiktos 8.3 priede. Išsamūs visų metaduomenų aprašymai ir pavyzdžiai pateikiami SCORM standarto mokymo objektų agregavimo apraše.

### 2.3.3. *MOKYMO OBJEKTO TURINIO PAKAVIMAS*

Išskirsime svarbiausius aspektus turinio pakavimo etape, kurie aptariami SCORM standarte.

SCORM reikalavimai paketo direktorijai:

- Paketo direktorija turi būti pavadinta mokymo objekto vardu;
- Katalogas sukuriamas toje pačioje direktorijoje, kurioje yra mokymo objektas.

SCORM paketą turi sudaryti šie failai:

- SCORM XML valdymo failai:
  - adlcp\_rootv1p2.xsd
  - imscp\_rootv1p1p2.xsd;
  - imsmd\_rootv1p2p1.xsd;
- HTML atvaizdavimo failai:
  - MO.htm;
  - tuscia.htm;
- SCORM Logika skriptas.

SCORM reikalavimai PIF failui:

- failas gali turėti PIF arba ZIP priesagą (pvz., 1\_pamoka.ZIP, 1\_pamoka.PIF);
- turi atitikti v2.04g archyvas, atitinkantis RFC 1951 reikalavimus.

## 2.4. **SISTEMOS REIKALAVIMŲ APIBENDRINIMAS**

Komponavimo sistema turi leisti nurodyti mokymo objekto failą ar visą katalogą su mokymo objektais. Vartotojas, išrinkdamas mokymo objektą, turi turėti galimybę atrinkti tik jį dominančių ob-

jektų tipus, pvz., rodyti tik HTML tipą. Sykiu vartotojas turi nesirūpinti su tuo mokymo objektu susijusiais objektais, jie turi būti išrenkami pačios sistemos.

Sistema pasirūpina teisių tikrinimu: jei atliekant kokį nors veiksmą teisės ribojamos, vartotojui yra pateikiamas klaidos pranešimas. Jei vartotojas, dirbdamas su programa ką nors užmiršo nurodyti, sistema jam turi pranešti, kas būtent buvo pamiršta.

Aprašant metaduomenis, sistema turi tikrinti, ar jie yra suderinti su SCORM standarte aprašytaisiais ir ar jų pakankamai nurodyta.

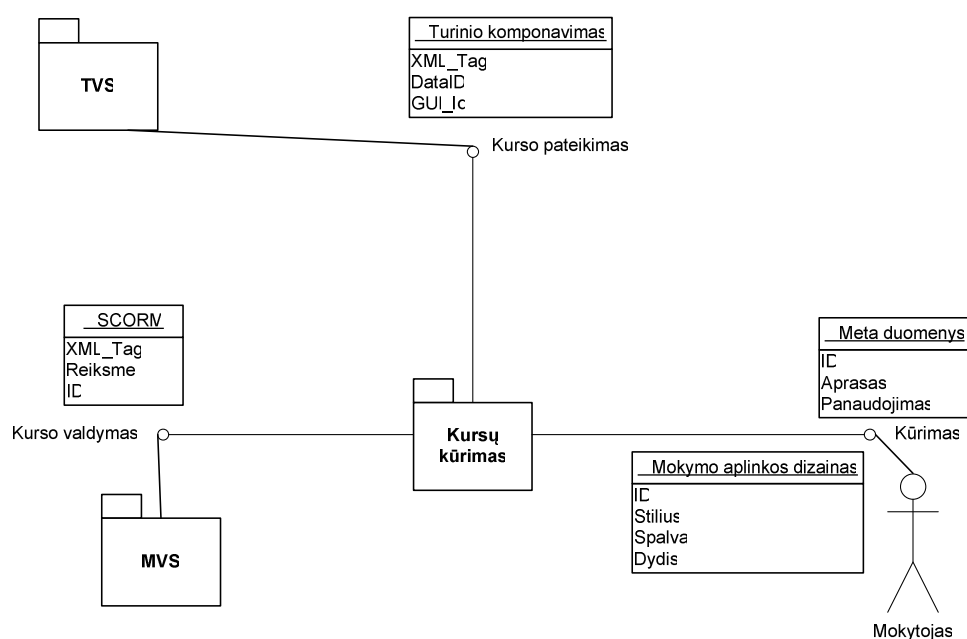
Apibrėžėme visus sistemai keliamus reikalavimus. Nurodėme visos sistemos elgsenos sekas - tiek sudėtingesnių etapų, tokių kaip metaduomenų aprašymo ir turinio pakavimo sekas. Jais naudodamiesi, sudarysime elektroninių kursų mokymo objektų komponavimo sistemos projektą. Pateiksime vartotojo sąsajos projektą.

## 2.5. DALYKINĖS SRITIES PRADINĖ PAKETINĖ ARCHITEKTŪRA

Šiame skyrelyje pabandysime apibrėžti, iš kokių modulių turėtų būti surinkta pilna elektroninių kursų mokymo objektų komponavimo sistema. Remdamiesi SCORM standartu, elektroninį mokymą išskaidome į šias sritis (pav. 2.9):

- Kurso pateikimas – dažniausiai realizuojamas kaip turinio valdymo sistema;
- Kurso valdymas - dažniausiai realizuojamas kaip mokymo valdymo sistema;
- Kurso kūrimas – dažniausiai realizuojamas kaip atskira vartotojo programa.

Kaip jau aptarėme reikalavimų specifikacijoje, nagrinėsime kurso kūrimo sritį, ją apribodami iki mokymo objektų komponavimo. Komponavimas, kaip minėjome, yra kurso kūrimo dalis, atmetus pačių mokymo objektų kūrimą ir koncentruojantis tik ties organizavimu, pateikimu, valdymu, aprašymu.

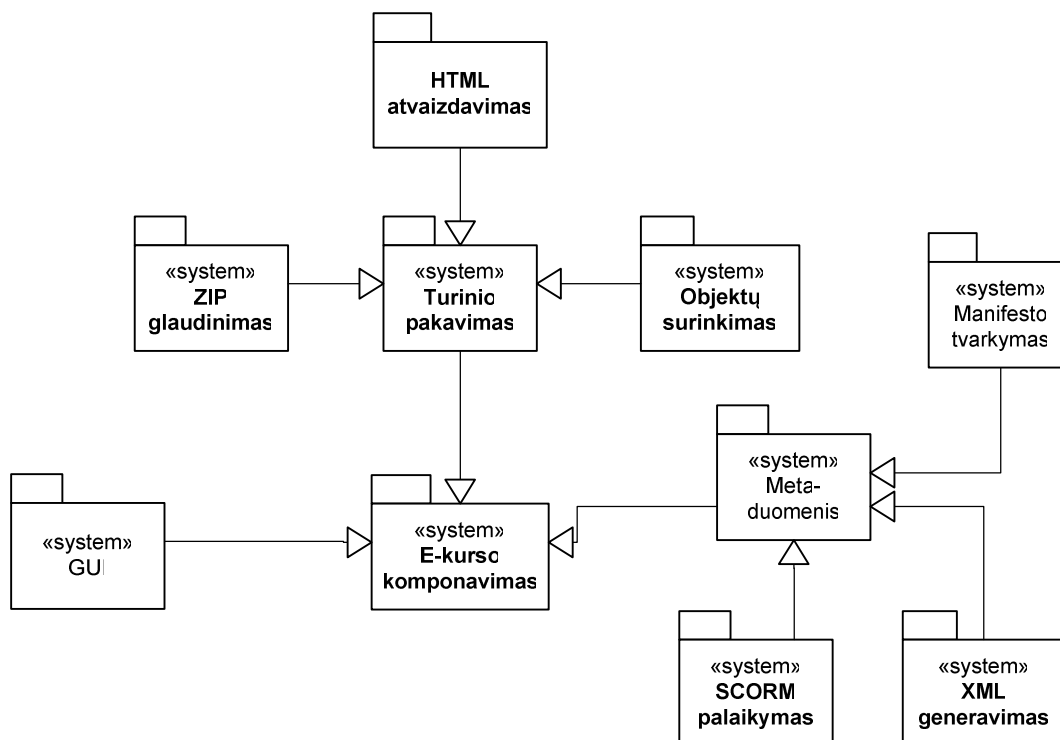


2.9 pav. Pradinė kursų kūrimo architektūra

## 2.6. KOMPONAVIMO SISTEMOS PAKETINĖ ARCHITEKTŪRA

Apibrėšime tiksliai, kokie komponentai sudarys būsimą mūsų komponavimo sistemą. Išskaidome kurso kūrimo sritį į smulkesnius paketus, kuriuos būtų patogu realizuoti. Kadangi, kaip anksčiau minėjome, darbe koncentruojamės į SCORM standarto problemas, todėl dar atmetame ir mokymo aplinkos dizaino ir mokymo objektų kūrimą, nes tai galima atlikti kitomis priemonėmis, skirtoms būtent tokiam darbui.

Kuriamą sistemą išskaidome į dvi pagrindines posistemes, kurias dar susmulkiname į konkrečias funkcijas, kad būtų lengviau specifikuoti ir realizuoti išskirtas funkcijas (pav. 19).



2.10 pav. Elektroninių kursų mokymo objektų komponavimo sistemos komponentinė schema;

2.2 lentelėje pateikiamas schemeje pavaizduotų paketų paaiškinimas. Toliau paanalizuosime smulkiau kiekvieną posistemę.

2.2 lentelė - Elektroninio kurso mokymo objektų komponavimo sistemos paketų paaiškinimas

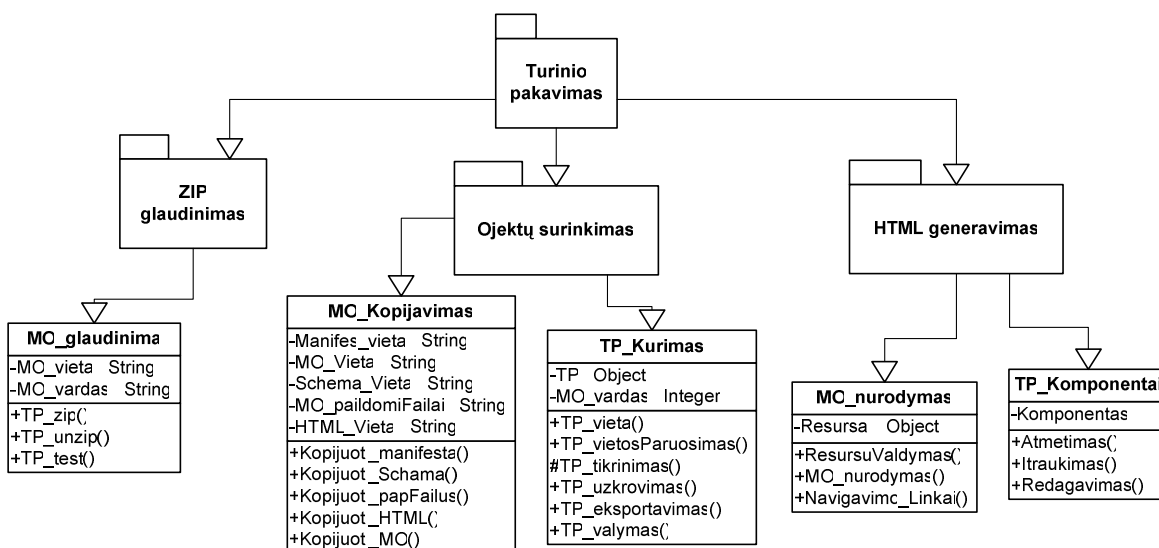
Posistemė	Komponentas	Aprašymas
Meta-duomenys	SCORM palaikymas	Užtikrina standarto SCORM palaikymą, tikrina duomenų teisingumą
	XML generavimas	Realizuoja XML struktūrų generavimą
	Manifesto tvarkymas	Manifesto failo tvarkymas
E-kurso komponavimas	ZIP glaudinimas	Mokymo objektų komponentų paketo PIF failo kūrimas
	Objektų surinkimas	Surenkami visi reikalingi mokymo objektai, schema failai ir kitos bylos, reikalingos paketo sukūrimui
	HTML atvaizdavimas	Mokymo objekto HTML atvaizdavimo kūrimas



### 2.6.1. TURINIO PAKAVIMO POSISTEMĖ

Ši posistemė atlieka mokymo objektų surinkimą, SCORM palaikymui reikalingų failų surinkimą į vieną vietą. Taip pat čia yra sugeneruojamas HTML failas, kuris leidžia atvaizduoti mokymo objektą. Šią posistemę dar išskaidome į konkrečias funkcijas atliekančius komponentus (pav. 20):

- Objektų surinkimas – jį sudaro dvi pagrindinės klasės:
  - TP\_kūrimas, skirtas vietos paruošimui, katalogo sukūrimui, kitiems paruošiamiesiems darbams.
  - MO\_Kopijavimas – apdoroja visas operacijas, susijusias su įvairių objektų surinkimu į vieną paketą.
- ZIP glaudinimas – pagrindinė funkcija atlikti sukurto mokymo objekto glaudinimą ZIP formatu, atitinkantį SCORM reikalavimus<sup>15</sup>.
- HTML generavimas – pagrindinė funkcija suformuoti HTML formą, kurios pagalba būtų atvaizduojamas mokymo objektas net ir be MVS. HTML generavimą sudaro dvi pagrindinės klasės: MO\_nurodymas ir TP\_Komponentai.

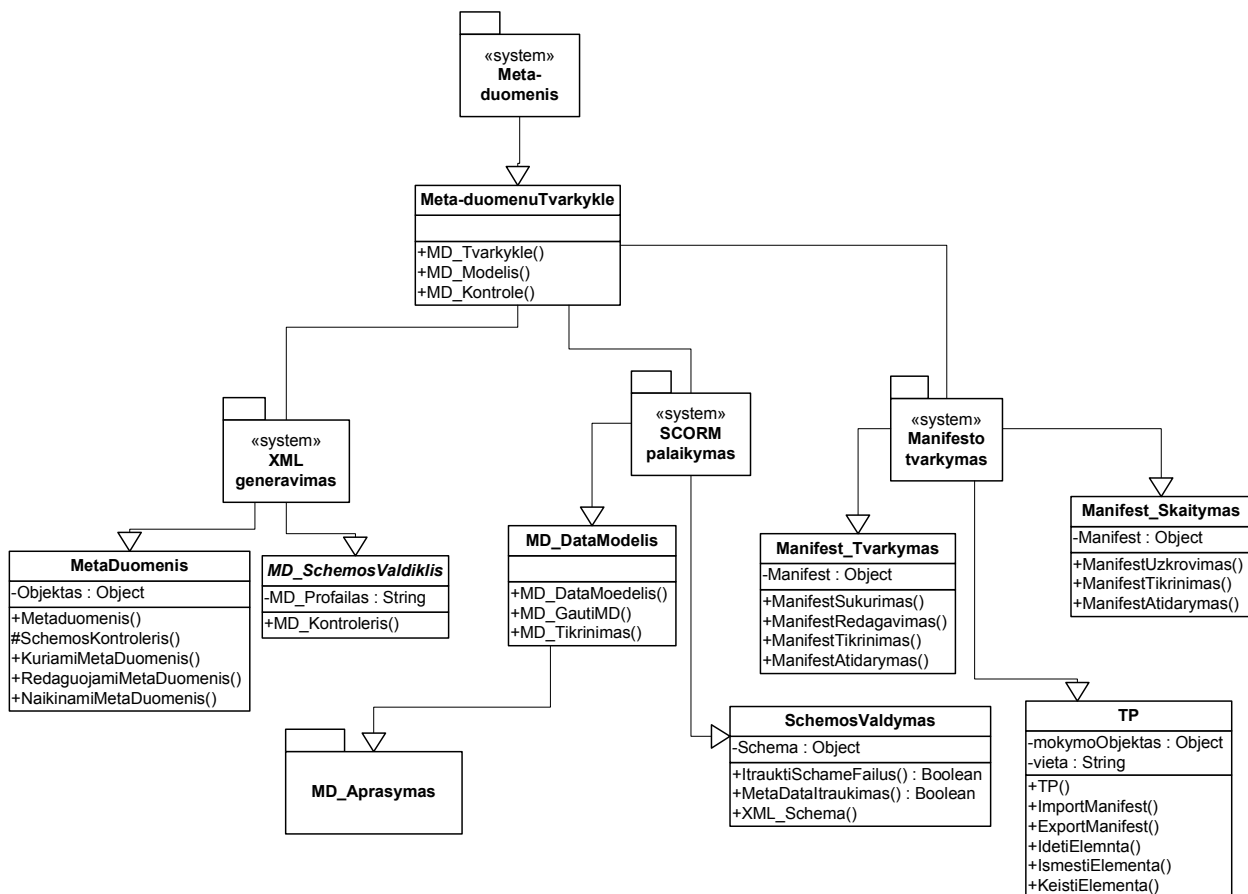


2.11 pav. Turinio pakavimo subsistemos išsami klasių diagrama

### 2.6.2. METADUOMENŲ APRAŠYMO POSISTEMĖ

Metaduomenų aprašymo posistemė tvarko viską, kas susiję su metaduomenimis, mokymo objekto aprašymu, generuoja XML failus, SCORM schemas. Aprašydami metaduomenis, remsimės SCORM standarto reikalavimais. Galime teigti, kad tai mūsų darbo svarbiausioji ir sudėtingiausioji dalis. Šioje posistemėje įsivėlus klaidai prarastume ir pačios sistemos funkcionalumą.

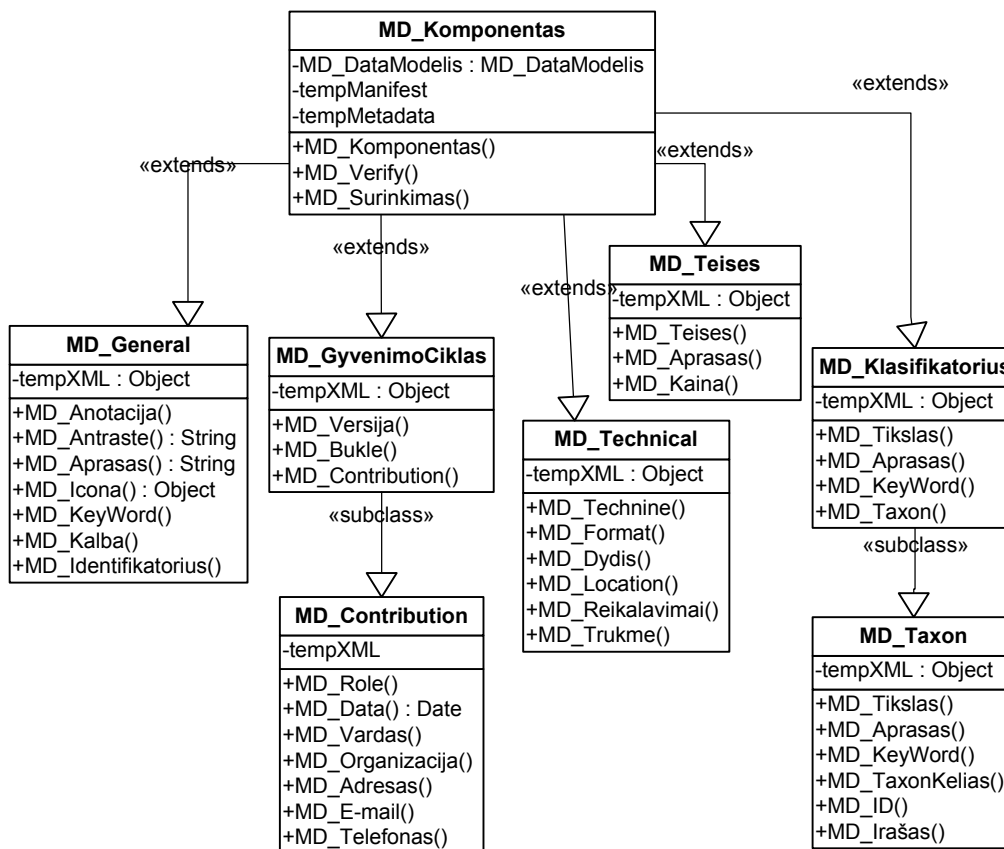
<sup>15</sup> SCORM Version 1.3 Working Draft 1 Content Agregation Model. Johnstown PA: ADL Technology Center. P.35.



2.12 pav. Meta-duomenų posistemės išsami struktūra

Posistemę MD\_Aprašymas, kurioje bus atliekamas vartotojui prieinamų metaduomenų įrašų keitimas, detalizuojame (pav. 2.10). Metaduomenų aprašymą remiantis SCORM standartu išskaidome į penkias grupes pagal konteinerius (žr. 8.3 priede):

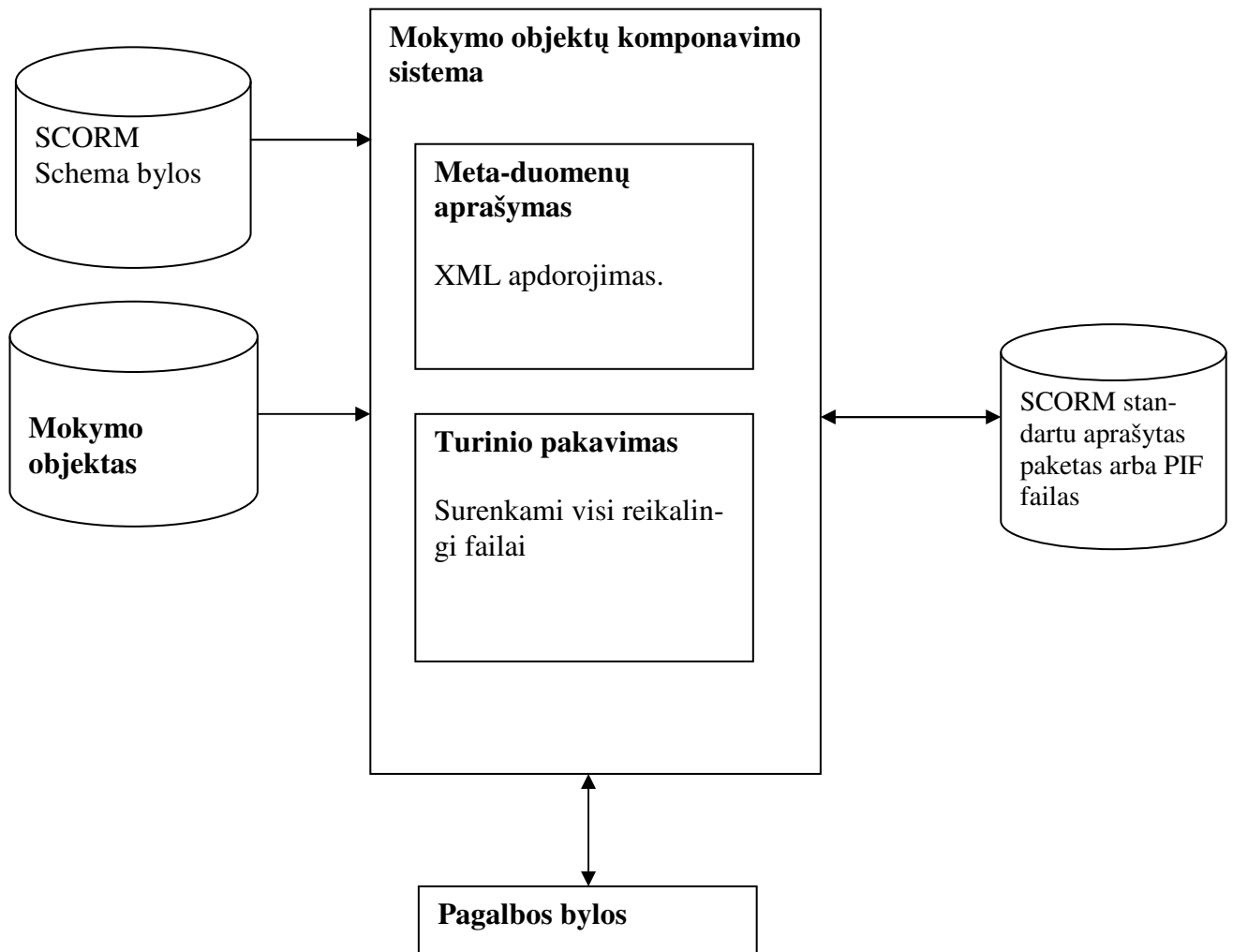
- MD\_General (<general>) – Aprašomi pagrindiniai metaduomenys. Vartotojui prieinami nurodymai: mokymo objekto pavadinimas (<title>), aprašas (<description>), raktiniai žodžiai (<keyword>) ir vartojama kalba (<language>).
- MD\_GyvenimoCiklas (<lyfeCycle>) – Aprašomi gyvavimo ciklo metaduomenys. Vartotojui prieinami nurodymai: mokymo objekto versija (<version>), būseną (<status>) ir kontribucija (<contribution>).
- MD\_Technical (<technical>) – Aprašomi techniniai metaduomenys. Vartotojui jokie nustatymai neprieinami.
- MD\_Taiesė (<rights>) – Aprašomo mokymo objekto apribojimai. Vartotojui prieinami nurodymai: mokymo objekto vertė (<cost>), naudojimo apribojimai (<copyrightandotherrestrictions>) ir paaiškinimas.
- MD\_Klasifikatorius (<classification>) – aprašomas mokymo objekto klasifikatorius. Vartotojui prieinami nurodymai: klasifikatoriaus paskirtis (<purpose>), aprašas (<description>), raktiniai žodžiai (<keyword>) ir pagal kokią sistemą yra klasifikuojama.



2.13 pav. MD\_Aprasymas posistemės išsami struktūra

## 2.7. REALIZACIJOS MODELIS

Schemoje 2.15 pavaizduotas elektroninių kursų mokymo objektų komponavimo sistemos realizacijos modelis. Sistemą, kaip minėjome, sudarys dvi posistemės: metaduomenų redaktorius ir turinio pakavimo posistemė. Prie pačios sistemos dar nurodome, kokie mums bus reikalingi išoriniai resursai ir ką turėtume gauti sistemos išėjime. Metaduomenų redaktoriaus posistemė su turinio redaktoriumi tiesiogiai nebus susijusios – jos dirbs lyg atskiros sistemos naudodamosi tik darbo rezultatais, jų net neanalizuojant. Kaip jau anksčiau minėjome, tai leistu sistemą lengviau pertvarkyti atsiradus naujai standarto versijai – reikėtų keisti tik metaduomenų redaktorių. Pakitus pakavimo mechanizmui, formatai ir panašiai – būtų keičiama tik pakavimo posistemė.



2.14 pav. Realizacijos modelis

## 2.8. ELEKTRONINIŲ KURSŲ KOMPONAVIMO SISTEMOS VARTOTOJO SĄSAJOS MODELIS

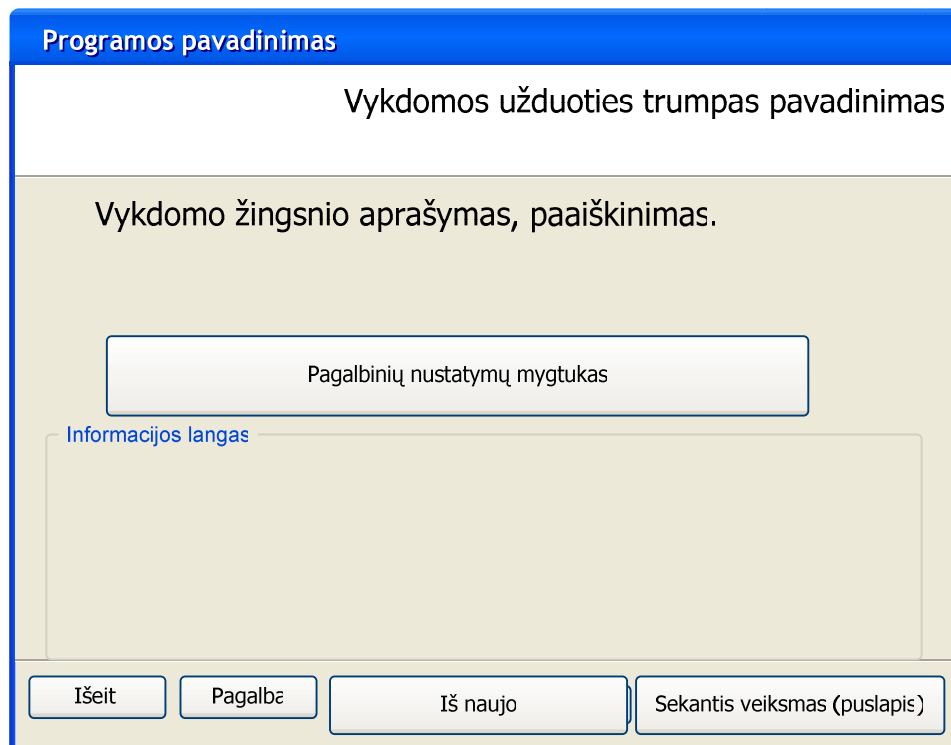
Reikalavimų dalyje nustatėme pagrindines užduotis, kurias programa turėtų atlikti. Atidžiau pa-nagrinėjus, matome, jog kiekvienos užduoties atlikimas yra tiesiogiai susijęs su tuo, ar prieš tai užduo-tis atlikta sėkmingai. Pavyzdžiui, norint aprašyti mokymo objekto metaduomenis, pirmiausiai mokymo objektą sėkmingai turėtume nurodyti. Dėl šios priežasties vartotojui veiksmai turėtų būti prienami tik patenkinus būtinas jo atlikimui sąlygas.

Principu, kai tik atlikus vieną veiksmą pereinama prie kito, veikia klasikinė vedlio (wizard) sąsa-ja. E-mokymo komponavimo sistemos vedlio projektas pateikiamas paveiksle 27.

Vedlio langą suskirstome į šias sritis:

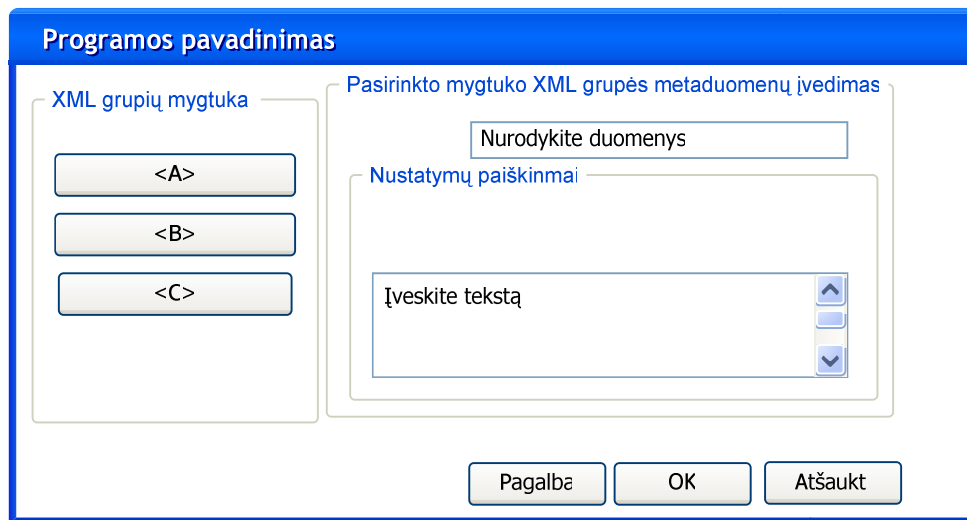
- Programos pavadinimas – šioje srityje visuose žingsniuose bus rodomas programos pavadi-nimas „Mokymo objektų komponavimo sistema“;
- Vykdomos užduoties trumpas aprašymas – trumpas, lakoniškas žingsnio, užduoties pavadi-nimas;

- Vykdomo žingsnio aprašymas – smulkus vykdomos užduoties aprašymas, mygtukų reikšmių paaiškinimas;
- Pagalbinių nustatymų mygtukas(ai) – žingsnio mygtukai, leidžiantys keisti einamos užduoties įvairius parametrus;
- Informacijos langas – vartotojui pateikiamos informacijos langas apie atliktus veiksmus;
- Standartiniai mygtukai:
  - Išėiti – baigiamas darbas su programa;
  - Pagalba – bendras darbo su programa aprašymas;
  - Iš naujo – vartotojui suklydus galima sugrįžti ir atlikti veiksmus iš naujo;
  - Sekantis puslapis – pereina į sekantį vedlio puslapį. Vartotojui turi būti prieinamas tik įvykdžius einamos užduoties reikalavimus.



**2.15 pav.** Vartotojo sąsajos projektas, parinis vedlio langas

Taip pat svarbi vartotojo sąsajos dalis yra mokymo objekto metaduomenų redagavimas. Anksčiau išskyrėme metaduomenų aprašymą į atskirą posistemę. Dėl įvairių niuansų vartotojo sąsaja taip pat turėtų būti adaptuota patogiam darbui su metaduomenimis. Kadangi aprašant metaduomenis nebe svarbu, kokia tvarka vartotojas juos aprašys, ar iš viso praleis, vedlio koncepcija netinka. Vartotojas, redaguodamas metaduomenis, turi turėti galimybę redaguoti norimus metaduomenis, neatsižvelgiant į eiliškumą. Tik vartotojui patvirtinus įvestus aprašus, jis turi būti informuotas apie metaduomenų suderinamumą su SCORM reikalavimais.



2.16 pav. Vartotojo sąsajos projektas metaduomenų aprašymui

## 2.9. PROJEKTAVIMO IŠVADOS

Šiame skyrelyje apibrėžėme ir identifikavome:

- Komponavimo sistemos veiklą;
- Sistemos elgseną;
- Patikslinome ir konkretizavome komponavimo sistemos reikalavimus;
- Išskyrėme reikalingiausias SCORM standarto aprašo reikalavimus.

Tuo remiantis suformavome ir sudarėme:

- elektroninių kursų mokymo objektų komponavimo sistemos paketinę architektūrą;
- schematinį realizacijos modelį;
- vartotojo sąsajos modelį.

Remdamiesi nuveiktais darbais, atlikome sistemos realizaciją su OpenScript programavimo kalba. Vartotojo sąsajai realizuoti pasinaudojome ToolBook Instructor įrankiu. Su sukurta komponavimo sistema vėliau atliksime eksperimentinį tyrimą, kuriuo bandysime įrodyti SCORM standarto praktinį panaudojimą ir pritaikymo naudingumą.

### **3. EKSPERIMENTINIS TYRIMAS**

#### **3.1. EKSPERIMENTO APRAŠYMAS**

Eksperimentą atliksime dviem požiūriais:

1. Testuosime sukurtos programos funkcionalumą, veikimą ir atitikimą reikalavimams;
2. Testuosime sukurtos programos sukomponuotą SCORM mokymo objektų aprašą įvairiose mokymo sistemose. Bandysime įvertinti šiuos aspektus:
  - a. elektroninio kurso įkėlimo testavimą;
  - b. elektroninio kurso atvaizdavimo teisingumą MVS;
  - c. SCORM aprašymo teisingumo vertinimą.

Eksperimentui naudosime specialiai tam sukurtą elektroninio mokymo interneto temų kursą. Kurso medžiaga paruošta kitomis programomis ir pateikiama pridedamame CD, kurso manifesto failas pateiktas 8.2 priede. Į patį mokymo turinį nesigilinsime, bet tuo pačiu stengsimės panaudoti kuo įvairnesnius mokymo objektų tipus, leidžiančius visapusiškai iširti sistemos patikimumą komponuojant mokymo objektus.

#### **3.2. ELEKTRONINIO KURSO STRUKTŪRA IR PARUOŠIMAS MOKYMO OBJEKTŲ KOMPONAVIMO SISTEMOS APDOROJIMUI**

Sukursime testavimo modelį taip, kad jis apimtų kuo daugiau mokymo objektų komponavimo sistemos funkcijų ir leistų patogiai ir objektyviai patikrinti elektroninio kurso suderinamumą su SCORM 1.2 standartu. Kurso išdėstymo hierarchija atitinka standartinę sukurtos sistemos organizaciją.

Sukuriame paprastą kursą iš pagrindinio puslapio ir keletą šalutinių kursų, plačiau aiškinančių temas. Pridedamame kompaktiniame diske pateikiamas testams naudotas mokymo objektų paketo PIF failas (1\_pamoka.zip).

Mokymo objektai ir kursas paruošiamas papildomomis priemonėmis, skirtomis teksto redagavimui ir paveiksliukų apdorojimui. Visas kursas pateikiamas mokymo objektais, suderintais su pagrindiniais interneto standartais. Mokymo objektai pateikiami dviem pagrindiniais formatais:

- HTML – pagrindiniai kurso struktūros mokymo objektai;
- JPG, PNG – pagalbinė vaizdinė mokomoji medžiaga.

Kursas sudarytas iš paprastos medžio tipo hierarchijos ir atitinka tokią schemą:

1. Pagrindinis puslapis, temų sąrašas – 1\_pamoka.html;
2. Pažintis su internetu – 1.htm;
  - image101.png;

- image102.jpg;
3. Asmeniniai kompiuteriai ir kompiuterių tinklai – 2.htm;
    - image101.png;
    - image102.jpg;
  4. Interneto paslaugos – 3.htm;
  5. Pagrindinės žiniatinklio sąvokos ir principai – 4.htm;
  6. Praktinė užduotis – uzd.doc.

### 3.3. SISTEMOS VEIKIMO EKSPERIMENTINIS TYRIMAS

Mokymo objektų komponavimo sistemą bandysime komponuoti įvairaus tipo mokymo objektus. Tirsime, kaip sistema susidoroja su įvairiu kiekiu susijusių failų, kaip reaguoja į kritines situacijas, kai tam tikroje vietoje negalima sukurti katalogo ar nurodytas mokymo objektas yra programai neprieinamas. Atlikti šie bandymai:

1. Paprasto mokymo objekto (reikia.txt) komponavimas.
2. Mokymo objekto (index.htm) komponavimas su priklausomais failais, kai kataloge index\_bylos vienas papildomas failas.
3. Mokymo objekto (index.htm) komponavimas su priklausomais failais, kai kataloge index\_bylos penki papildomi failai.
4. Mokymo objekto (index.htm) komponavimas su priklausomais failais, kai kataloge index\_bylos dvidešimt papildomų failų.
5. Sistemos elgesio tyrimas nurodant sistemai negalimų formatų failus, pvz. index.zip.
6. Mokymo objekto (index.htm) komponavimas su priklausomais failais, kai kataloge index\_bylos papildomas failas didesnis kaip 5Mb.
7. Mokymo kurso komponavimas ir paruošimas testavimui mokymo valdymo sistemose.

**3.1 lentelė** – Elektroninių kursų mokymo objektų komponavimo sistemos funkcionalumo tyrimas

Eil.nr.	Komponuojamą objektų	Išvada
1.	Reikia.txt	Darbas atliekamas sklandžiai, greitai. Jokių trukdžių nepastebėta
2.	Index.htm + 1 MO	Darbas atliekamas sklandžiai, greitai. Jokių trukdžių nepastebėta
3.	Index.htm + 5 MO	Darbas atliekamas sklandžiai, greitai. Jokių trukdžių nepastebėta
4.	Index.htm + 20 MO	Darbas atliekamas sklandžiai, greitai. Pastebimas nedidelis darbo sulėtėjimas, dėl interpretatoriaus neuron.exe naudojimo ir ne visiškai optimizuoto programos kodo.
5.	index.zip	Sistema parodo klaidą, jog šio failo negalima įtraukti.



6.	Index.htm + 5MB MO	Sistemos darbui mokymo objekto dydis neturi reikšmės
7.	Testinio kurso kūrimas	Darbas atliekamas sklandžiai, greitai. Jokių trukdžių nepastebėta. Peržiūrėjus manifesto failą SCORM aprašo pažeidimų nerasta.

Reziumuojant galima teigti, jog sistema pakankamai gerai susitvarko su užduotimi, kai elektroninis kursas nedidelis, jame nėra daug papildomų mokymo objektų. Didėjant mokymo objektų kiekiui, sistemos darbo greitis mažėja. Taip yra dėl papildomos OpenScript kalbos interpretatorių naudojimo ir nevisiškai optimizuoto programos kodo.

### 3.4. EKSPERIMENTINIO ELEKTRONINIO KURSO TYRIMAS

Kurso suderinamumą testuosime mokymo valdymo sistemose, kurios palaiko bent jau SCORM 1.2 versiją. Kaip jau anksčiau minėjome analizės dalyje, realizuodami sistemą testams naudojome dokebosLMS2004. Norėdami gauti objektyvius eksperimento rezultatus, atrinkome dar keletą MVS. Naudosime vien atviro kodo sistemas, nes jas lengva gauti, tvarkyti, tirti.

Vadovaudamiesi prieinamumo ir suderinamumo su SCORM 1.2 standartu, atrinkome šias MVS:

- dokebosLMS2004;
- ATutor 1.5.2;
- Dokeos 1.6.3;

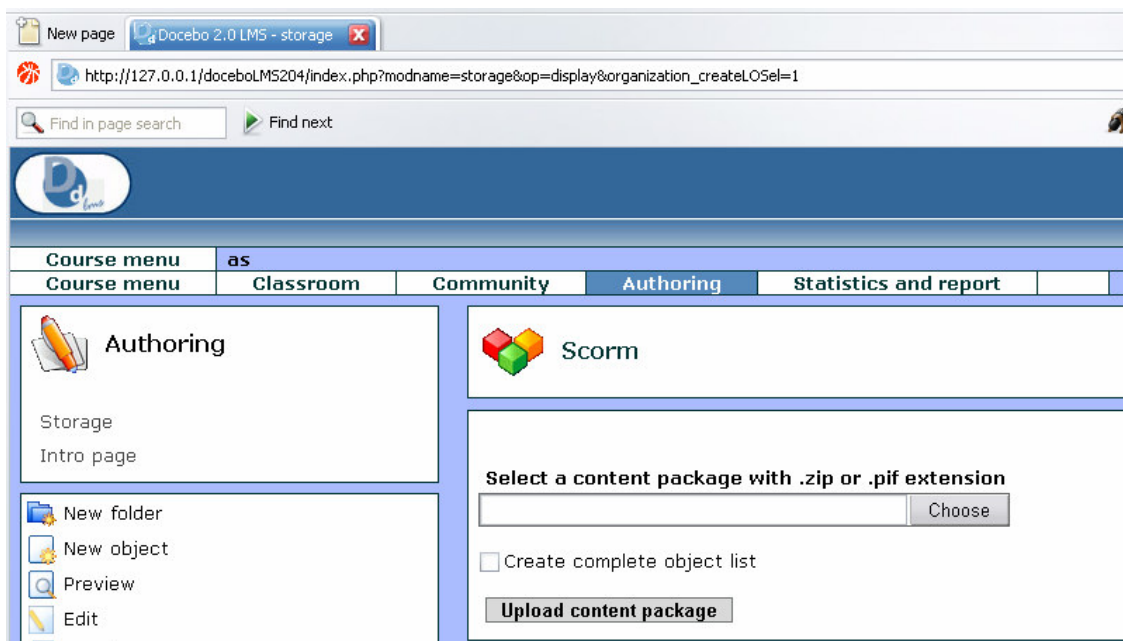
Praėjus metams, peržvelgus atviro kodo mokymo valdymo sistemas, pateikiamas [www.opensourcecms.com/](http://www.opensourcecms.com/), pastebimas akivaizdus SCORM standarto populiarėjimas. Jei prieš metus SCORM 1.2 standartą pilnai palaikė tik dokebos2004LMS, tai dabar tai atlieka kone visos MVS, tik kai kuriose dar nerealizuotos visos standarto numatytos funkcijos.

Eksperimento metu elektroninio mokymo sistemos buvo įdiegiamos Alache serveryje su PHP ir MySQL programine įranga. Kiekviena MVS buvo įdiegta pagal rekomenduojamas nuostatas su standartinėmis sistemos funkcijomis. ATutor 1.5.2 papildomai įdiegtas SCORM modulis.

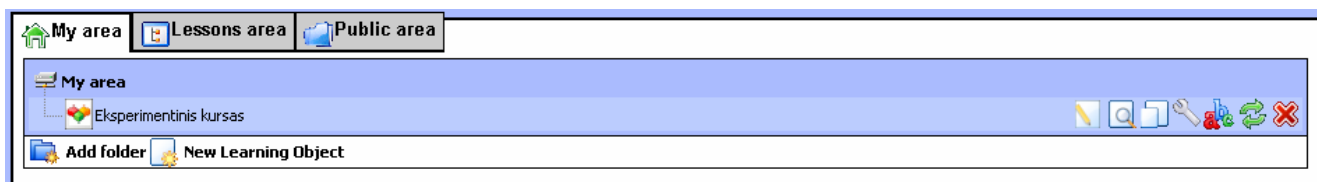
#### 3.4.1. ELEKTRONINIO KURSO ĮKĖLIMO TESTAVIMAS

Testavome kiekvienos mokymo valdymo sistemos mūsų sukurto kurso paketo 1\_pamoka.zip įkėlimo funkcionalumą, teisingumą.

- dokebosLMS2004 – kursas įkeliamas nurodžius PIF failą (3.1 pav.) be jokių klaidos pranešimų. Kursas atsiranda sistemos modulių sąrašė (3.2 pav.).

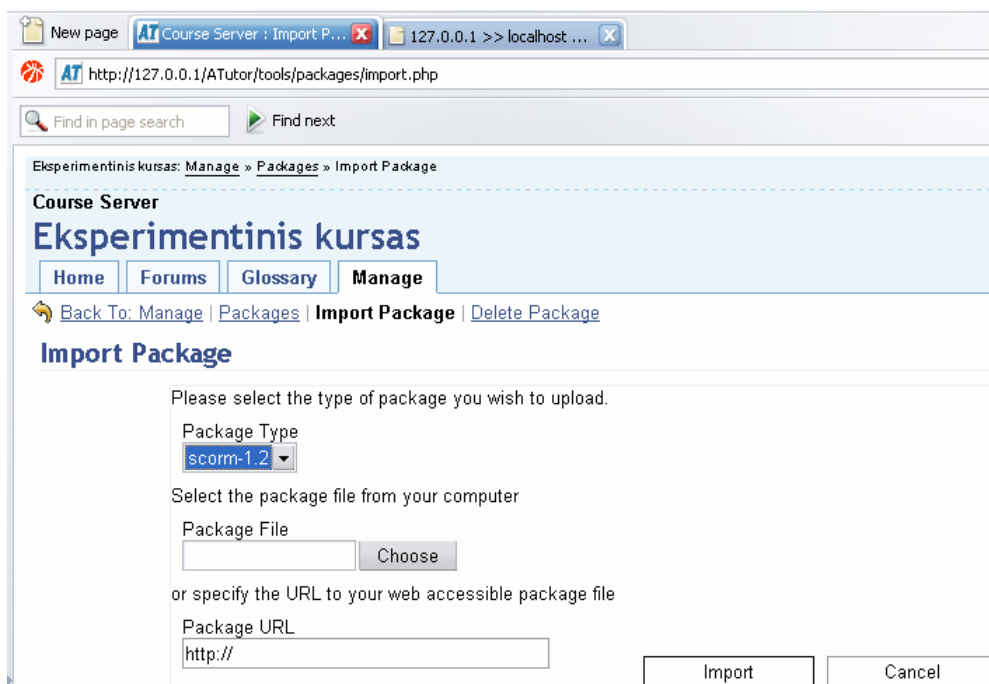


3.1 pav. dokebosLMS2004 kurso įkėlimo langas



3.2 pav. dokebosLMS2004 sistemoje įkeltas eksperimentinis kursas

- ATutor 1.5.2 - kursas įkeliamas nurodžius PIF failą (3.3 pav.). Įkėlus kursą sistema parodo pranešimą (3.4 pav.), jog kursas įkeltas sėkmingai. Kursas atsiranda sistemos modulių sąrašė.



3.3 pav. ATutor 1.5.2 kurso įkėlimo langas

## Packages

The package "Content" was imported successfully.

3.4 pav. ATutor 1.5.2. pranešimas po kurso įkėlimo

- Dokeos 1.6.3 - kursas įkeliamas nurodžius PIF failą (3.3). Įkėlus kursą sistema parodo pranešimą (3.5 pav.), jog kursas įkeltas sėkmingai.

My portal - My Organisation  
Doe John : My course list | My profile | My agenda | Platform Administration  
My portal > Current course > Learning Path

### Learning Path

[Add introduction text](#)

Upload IMS or Scorm Learning Path zip course :

The learning path was succesfully edited

[Create a Dokeos Learning Path](#) [Create a directory](#)

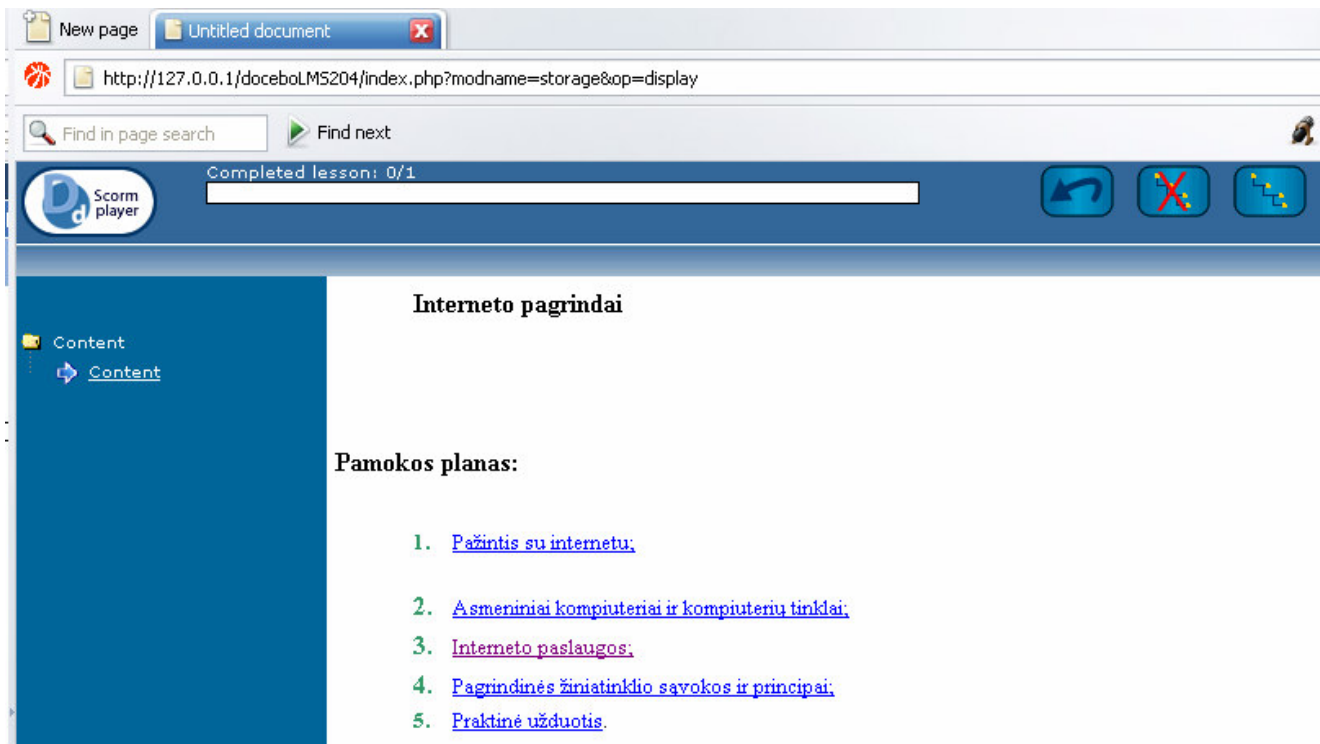
Name	Description
<a href="#">Eksperimentinis kursas</a>	

3.5 pav. dokeos 1.6.3 kurso įkėlimo langas

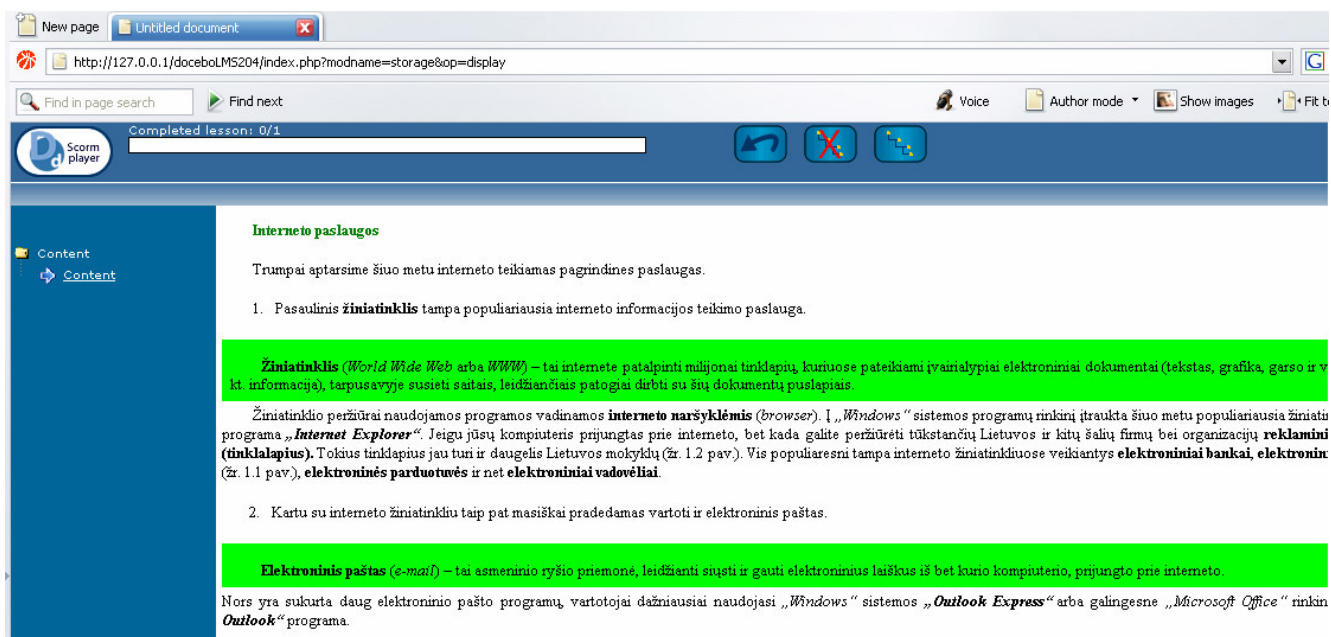
### 3.4.2. ELEKTRONINIO KURSO ATVAIZDAVIMO TEISINGUMO TYRIMAS

Testavome kiekvienos mokymo valdymo sistemos mūsų sukurto kurso paketo 1\_pamoka.zip atvaizdavimo funkcionalumą, teisingumą. Nagrinėjome pagrindinį puslapį (1\_pamoka.htm) ir interneto paslaugų temą (2.htm).

- dokebosLMS2004 – pagrindinis puslapis atvaizduojamas gerai (3.6 pav.), visos nuorodos veikia be jokių klaidos pranešimų. Turinys atvaizduojamas korektiškai (3.7 pav.), be jokių iškreipymų. Kalba atpažįstama teisingai.

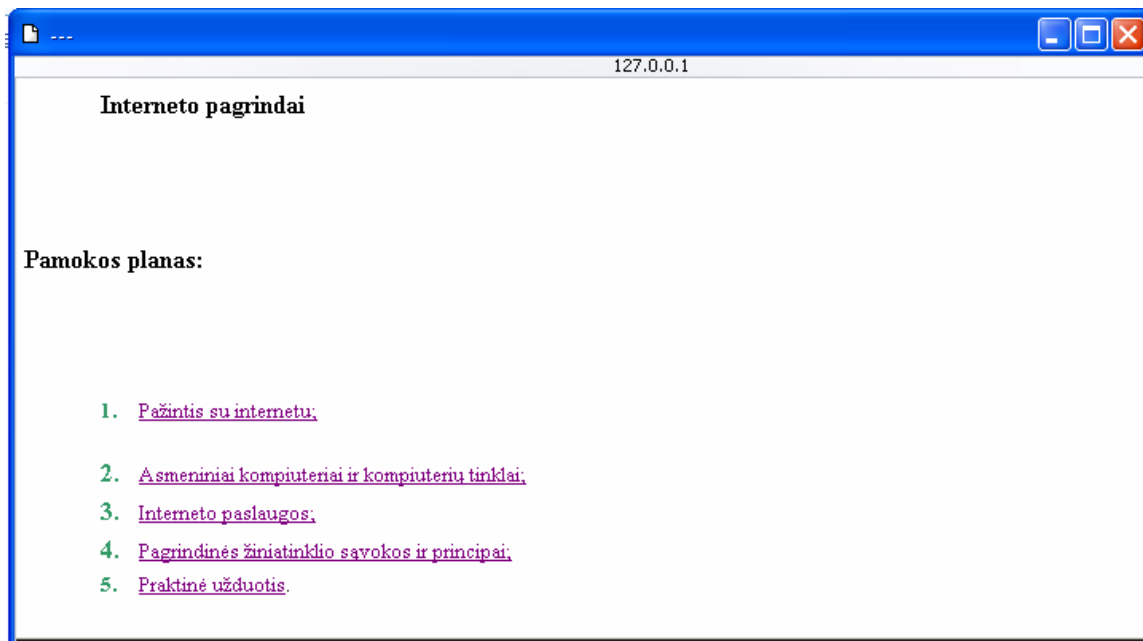


3.6 pav. docebosLMS2004 pagrindinio puslapio atvaizdavimas

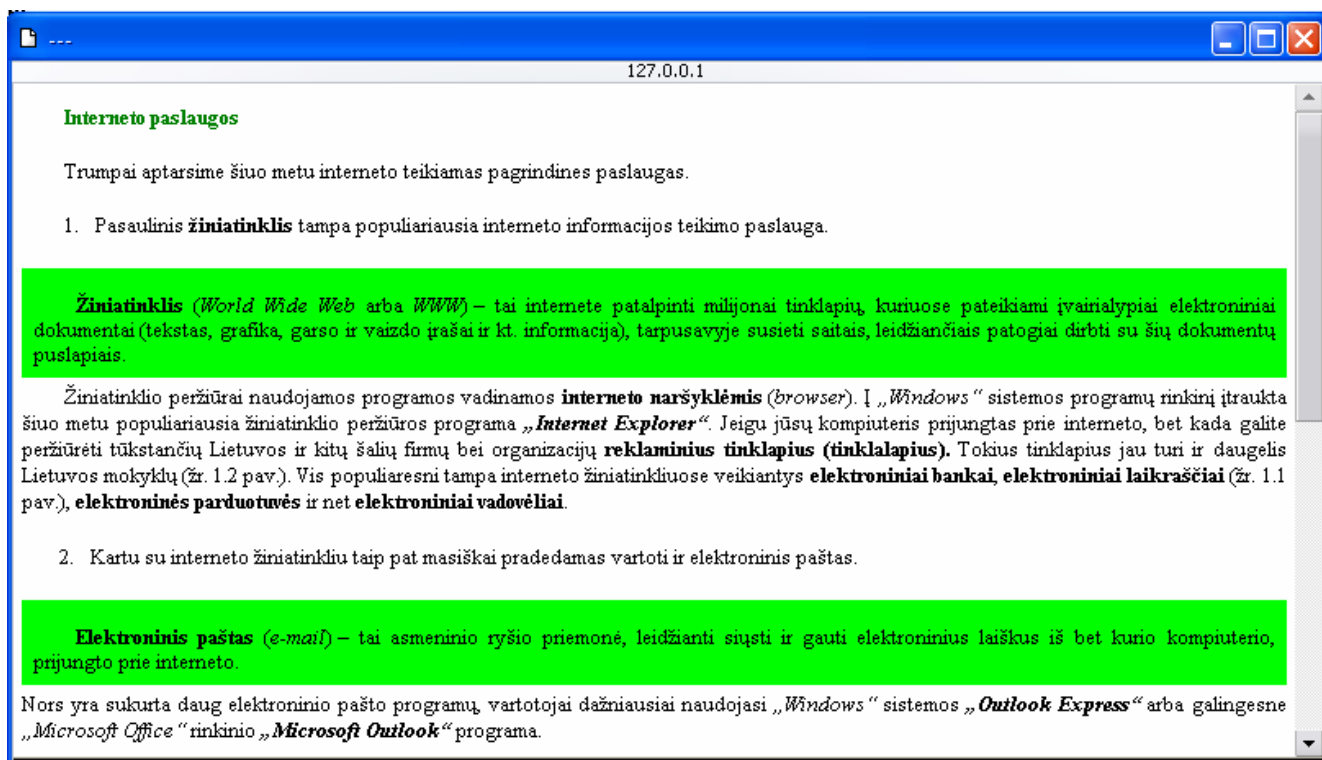


3.7 pav. docebosLMS2004 turinio vaizdavimas

- ATutor – pagrindinis puslapis atvaizduojamas atskirame lange (3.8 pav.), visos nuorodos veikia be jokių klaidos pranešimų. Turinys atvaizduojamas korektiškai (3.7 pav.), be jokių iškreipimų. Kalba atpažįstama teisingai.



3.8 pav. ATutor pagrindinio puslapio atvaizdavimas



3.9 pav. ATutor turinio vaizdavimas

- Dokeos – pagrindinis puslapis atvaizduojamas gerai (3.10 pav.), visos nuorodos veikia, papildomi objektai atpažįstami gerai, be jokių klaidos pranešimų. Turinys atvaizduojamas teisingai (3.11 pav.), su nedideliais iškreipymais. Kalba atpažįstama teisingai.

The screenshot shows a web browser window with a single tab titled 'Learning Path - Dokeos'. The address bar contains the URL: `http://127.0.0.1/dokeos-1.6.3/clarine/scorm/showinframes.php?openfir...`. The page header includes 'My portal - My Organisation' and 'Eksperimentinis kursas CL - Mokytojas'. Below the header, there are navigation links: 'Doe John : My course list | My profile | My agenda | Platform Administration' and a 'Logout' button. A breadcrumb trail reads: 'My portal > Current course > Learning Path > 1\_pamoka\_PIF'. The main content area is titled 'Interneto pagrindai' and features a 'Content' section with a progress indicator 'Completed 0 %'. Below this, a 'Pamokos planas:' (Lesson plan) is listed with five numbered items:

1. [Pažintis su internetu;](#)
2. [Asmeniniai kompiuteriai ir kompiuterių tinklai;](#)
3. [Interneto paslaugos;](#)
4. [Pagrindinės žiniatinklio sąvokos ir principai;](#)
5. [Praktinė užduotis.](#)

At the bottom of the content area, there are navigation icons for back, forward, and search.

3.10 pav. Dokeos pagrindinio puslapio atvaizdavimas

This screenshot shows the same Dokeos interface but with the 'Interneto paslaugos' (Internet services) section expanded. The progress indicator remains at 'Completed 0 %'. The content area is titled 'Interneto paslaugos' and contains the following text:

Trumpai aptarsime šiuo metu interneto teikiamas pagrindines paslaugas.

1. Pasaulinis **žiniatinklis** tampa populiariausia interneto informacijos teikimo paslauga.

**Žiniatinklis** (*World Wide Web* arba *WWW*) – tai internete patalpinti milijonai tinklapių, kuriuose pateikiami įvairiaypiai elektroniniai dokumentai (tekstas, grafika, garso ir vaizdo informacija), tarpusavyje susieti saitais, leidžiančiais patogiai dirbti su šių dokumentų puslapiais.

Žiniatinklio peržiūrai naudojamos programos vadinamos **internetu naršyklėmis** (*browser*). Į „Windows“ sistemos programų rinkinį įtraukta šiuo metu populiariausia žiniatinklio programa „**Internet Explorer**“. Jeigu jūsų kompiuteris prijungtas prie interneto, bet kada galite peržiūrėti tūkstančių Lietuvos ir kitų šalių firmų bei organizacijų **reklaminius (tinklapius)**. Tokius tinklapius jau turi ir daugelis Lietuvos mokyklų (žr. 1.2 pav.). Vis populiaresnis tampa interneto žiniatinkliuose veiksiantys **elektroniniai bankai, elektroniniai** (1.1 pav.), **elektroninės parduotuvės** ir net **elektroniniai vadovėliai**.

2. Kartu su interneto žiniatinkliu taip pat masiškai pradėdama vartoti ir elektroninis paštas.

**Elektroninis paštas** (*e-mail*) – tai asmeninio ryšio priemonė, leidžianti siųsti ir gauti elektroniškus laiškus iš bet kuro kompiuterio, prijungto prie interneto.

Nors yra sukurta daug elektroninio pašto programų, vartotojai dažniausiai naudojami „Windows“ sistemos „**Outlook Express**“ arba galingesnė „**Microsoft Office**“ rinkinį „**Outlook**“ programa.

3.11 pav. Dokeos turinio vaizdavimas

### 3.4.3. SCORM APRAŠYMO TEISINGUMO ANALIZĖ

Atlikus elektroninio eksperimentinio mokymo kurso tyrimą dviem etapais nustatyta:

- eksperimentinis mokymo kursas visose MVS sistemose įkeliamas be klaidos pranešimo;
- MVS kurso mokymo objektai atpažįstami korektiškai;
- mokymo objektai atvaizduojami visose bandytose MVS praktiškai vienodai su nežymiais skirtumais dėl SCORM valdiklio skirtingų realizacijų.

Todėl galime daryti išvadą, jog:

- kurso pakavimas atitinka visus SCORM 1.2 reikalavimus;
- XML struktūros atitinka SCORM 1.2 reikalavimus.

### 3.5. EKSPERIMENTO APIBENDRINIMAS (IŠVADOS)

Atlikus eksperimentą, galime teigti, jog:

- mokymo objekto aprašymas visiškai atskirtas nuo konkrečios MVS;
- SCORM standartas leidžia funkcionaliai ir aiškiai aprašyti mokymo objektus;
- elektroninių kursų mokymo objektų komponavimo sistema pakankamai gerai atlieka mokymo objektų komponavimą.

Sukurtą sistemą, galima taikyti mokymo objektų pateikimui mokymo valdymo sistemoms sudėtingoms su SCORM 1.2 standartu. Norint sistemą padaryti funkcionalesnę, reikėtų didinti vartotojo įtaką metaduomenų aprašymui, leisti vartotojui komponuoti daugiau mokymo objektų vienu metu, kurti sudėtingesnes hierarchines struktūras.

## 4. VARTOTOJO DOKUMENTACIJA

### 4.1. PROGRAMOS APRAŠAS

Realizuotas sistemos prototipas demonstruoja vieną iš galimų vartotojo ir techninio sprendimo įgyvendinimo būdų. Pagrindinis tokios sistemos tikslas – įrodyti SCORM standarto naudingumą, panaudojamumą ir privalumus. Visa tai mes aptarėme eksperimentinėje dalyje, kai lyginome mūsų sukurta interneto temų e-mokymo kursą keliose MVS.

Programa veikia vedlio principu.

### 4.2. VARTOTOJO SĄSAJA

Programos vartotojo sąsaja paremta vedlio principu. Kiekvienas darbas išskaidytas į žingsnius-ekranus. Tik atlikus nurodytus darbus galima pereiti į sekantį žingsnį. Kai kuriuose žingsniuose nėra jokių nustatymų, juose tiesiog aprašomi tuo metu programoje atliekami veiksmai.

#### 4.2.1. PAGRINDINIULOSE ŽINGSNIULOSE ESANČIŲ MYGTUKŲ APRAŠYMAS

Pagrindiniai mygtukai:

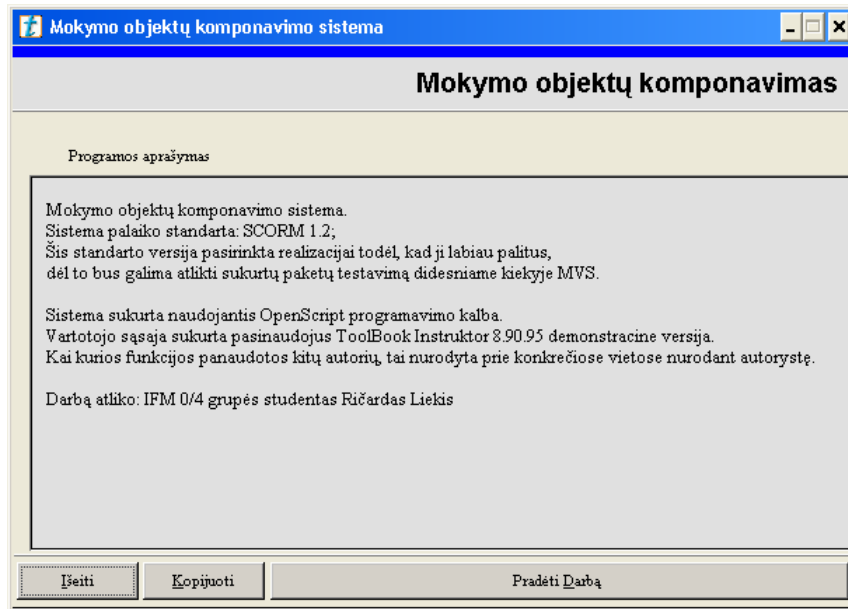
- *Išeiti* – paspaudus ši mygtuką baigiamas darbas su programa;
- *Kopijuoti* – leidžia nukopijuoti programos aprašymo tekstą į mainų atmintį;
- *Pradėti darbą* – pradedamas darbas su programa;
- *Pagalba* – paspaudus ši mygtuką vartotojui pateikiamas trumpas aprašymas, kaip naudotis programa;
- *Nurodykite mokymo objekto vietą* – nurodome mokymo objekto vietą diske;
- *Toliau* – perėjimas iš vieno žingsnio į kitą. Prieinamas tik tada, kai įvykdytos to žingsnio būtinos sąlygos;
- *Iš naujo* – leidžia sugrįžti prie pradinio žingsnio – mokymo objekto išrinkimo;
- *Redaguoti mokymo objekto duomenis* – paspaudus ši mygtuką, pereinama prie mokymo objekto metaduomenų redagavimo;
- *Kurti SCORM paketą* – sukuriamas SKORM paketas, surenkami visi reikalingi objektai, failai;
- *Peržiūrėti sukurto paketo manifesto failą* – atidaro sukurtą imsmanifest.xml failą peržiūrai (atidaroma su programa, kuriai sistemoje priskirti XML failai);
- *Sukurto paketo suspaudimas ZIP formatu* – leidžia sukurti PIF failą, kaip numatyta SCORM specifikacijoje.



#### 4.2.2. PROGRAMĄ SUDARANTYS PAGRINDINIAI ŽINGSNIAI

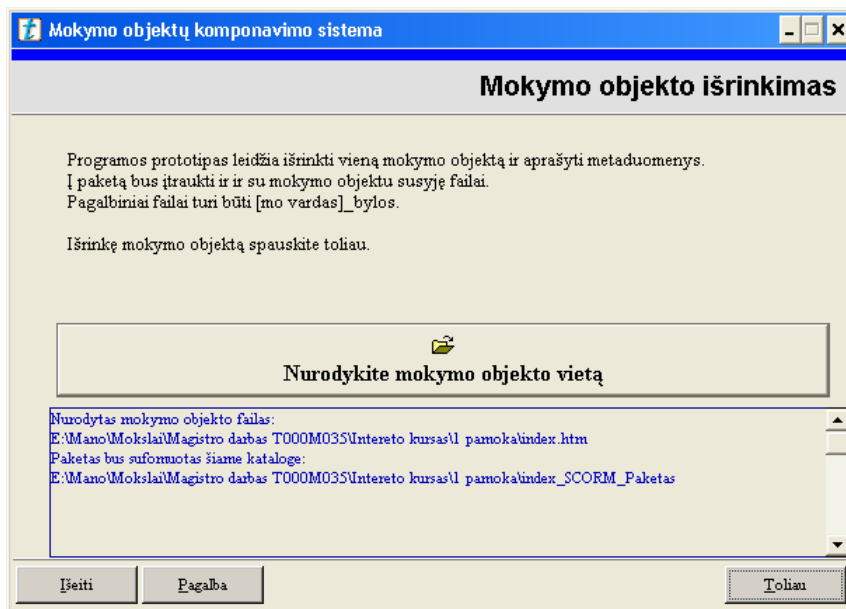
Programa sudaryta iš šių pagrindinių žingsnių (mygtukų reikšmes žiūrėkite skyrelyje „Pagrindiniuose žingsniuose esančių mygtukų aprašymas“):

1. Apie programą – pradžios langas paleidus sistemą, kuriame nurodomi kūrėjai ir trumpas bendras aprašas apie programą.



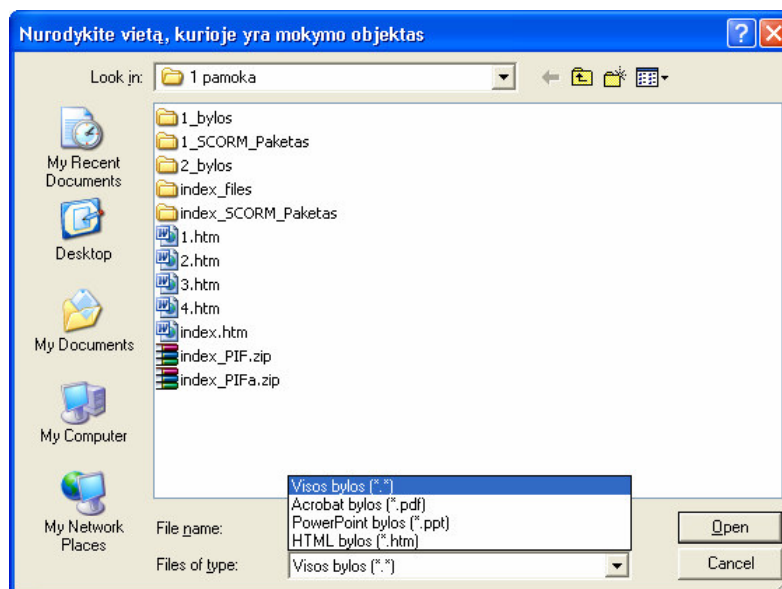
4.1 pav. 1 žingsnis: Apie programą;

2. Mokymo objekto išrinkimas – šiame žingsnyje vartotojas turi išsirinkti norimą mokymo objektą (asset file), kurį nori įtraukti į paketą.

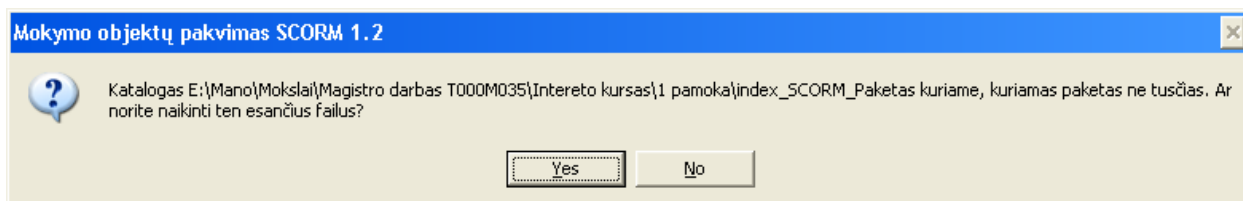


4.2 pav. 2 žingsnis: mokymo objekto išrinkimas

Vartotojas, pasirinkdamas mokymo objektą, gali nurodyti filtrus, dėl ko greičiau bus rastas norimas failas.

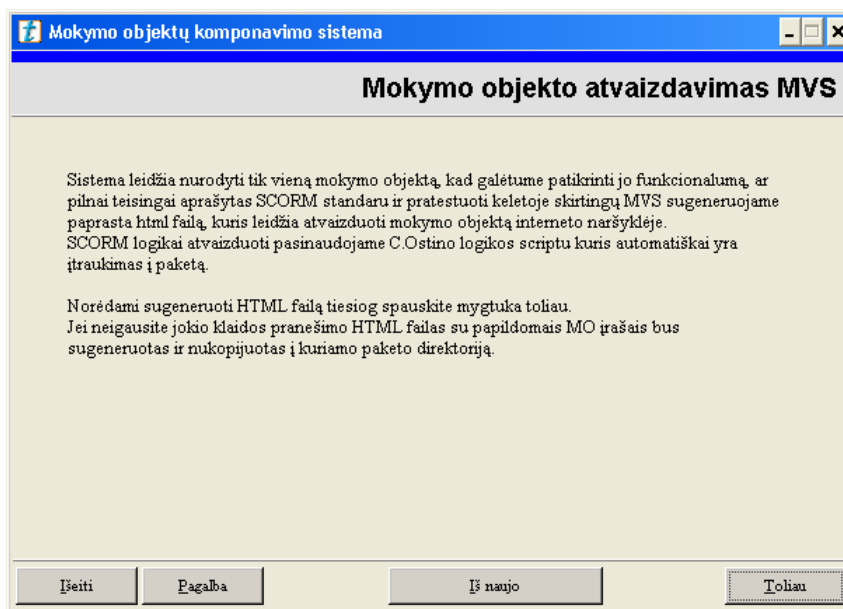


4.3 pav. Mokymo objekto vietos nurodymo langas ir galimi filtrai



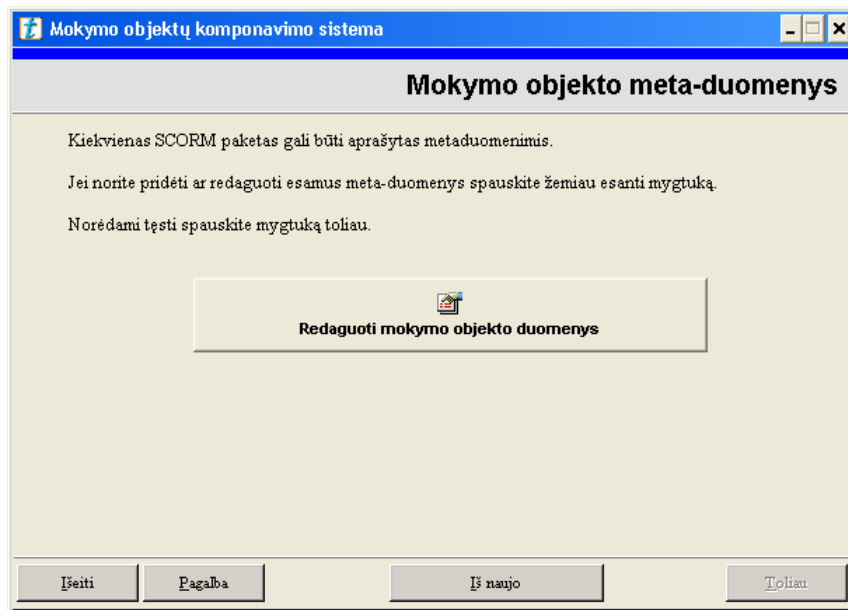
4.4 pav. Vartotojo perspėjimas, jei toks katalogas jau egzistuoja.

- Šablono parinkimas mokymo objektų atvaizdavimui MVS – sugeneruojamas HTML failas, leidžiantis atvaizduoti mokymo objektą.



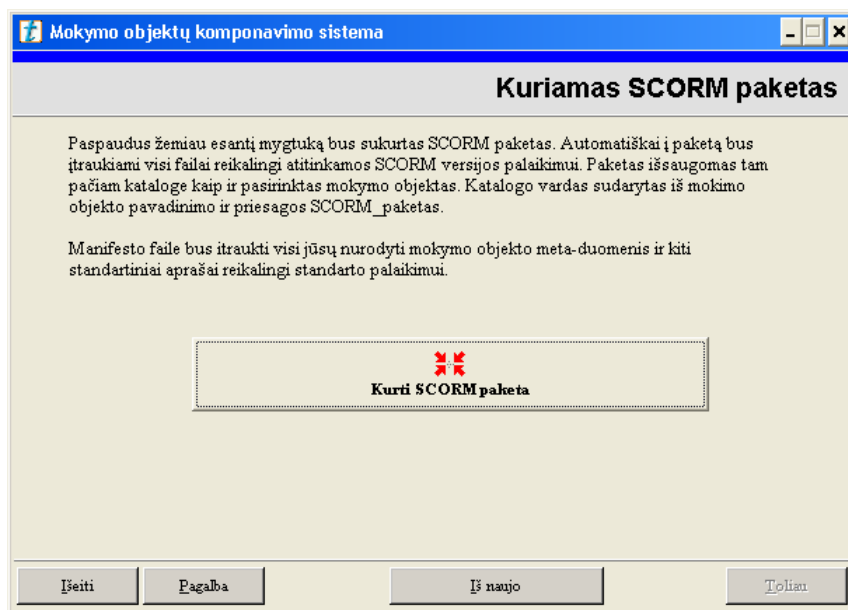
4.5 pav. 3 žingsnis: Mokymo objekto atvaizdavimas

- Mokymo objekto meta-duomenys – šiame žingsnyje galime išsikviesti metaduomenis aprašančią posistemę. Metaduomenų posistemės aprašymas pateikiamas skyrelyje „*Metaduomenų aprašymo posistemė*“.



4.6 pav. 4 žingsnis: Mokymo objekto meta-duomenys

5. Kuriamas SCORM paketas – šiame žingsnyje sukuriamas pilnavertis SCORM paketas iš visų reikalingų failų. Nepavykus to padaryti, vartotojas apie tai yra informuojamas atitinkamu klaidos pranešimu. Sėkmės atveju tiesiogiai pereinama į paskutinį žingsnį.



4.7 pav. 5 žingsnis: Kuriamas SCORM paketas

6. Baigiamasis žingsnis, kuriame galime peržiūrėti manifesto failą ir atlikti paketo suspaudimą ZIP formatu (skurti PIF failą). Lango apačioje pateikiama informacija apie sukurtą paketą.

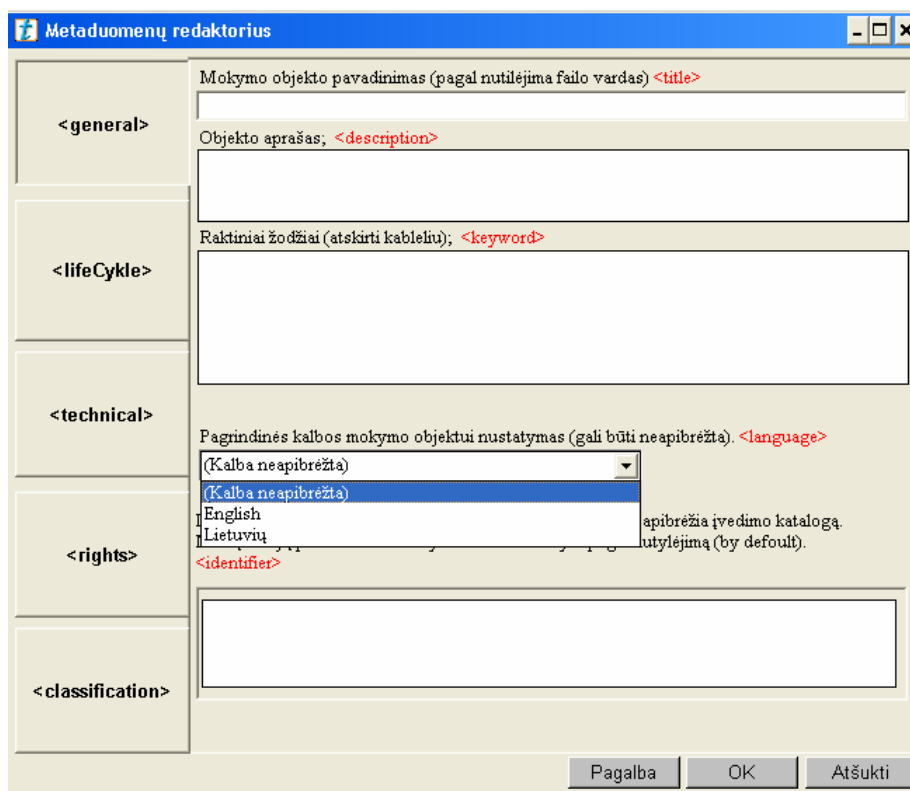


4.8 pav. 6 žingsnis: Baigiamas žingsnis

#### 4.2.3. METADUOMENŲ APRAŠYMO POSISTEMĖ

Metaduomenų posistemėje mygtukus suskirstome pagal apimančius tagus.

- <general> - vartotojas nurodo pagrindinius duomenis apie mokymo objektą;



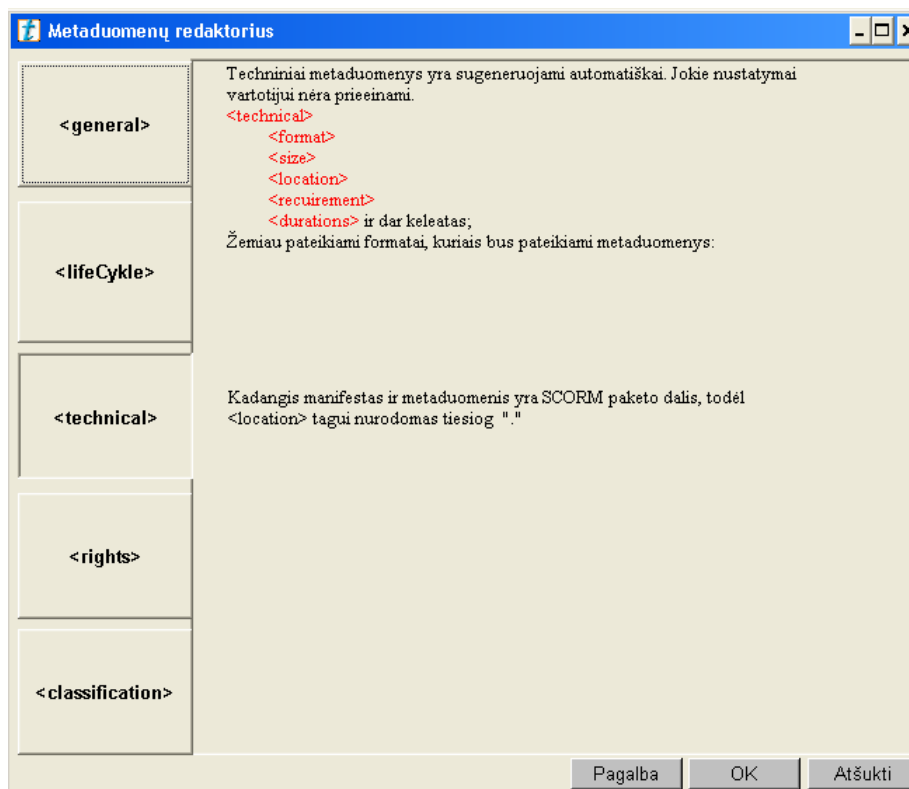
4.9 pav. Pagrindinių &lt;general&gt; metaduomenų aprašymas

- <lyfeCycle> - vartotojas nurodo mokymo objekto kūrimo etapą ir su tuos susijusius meta-  
duomenys

4.10 pav. Gyvavimo ciklo <lifeCycle> aprašymas

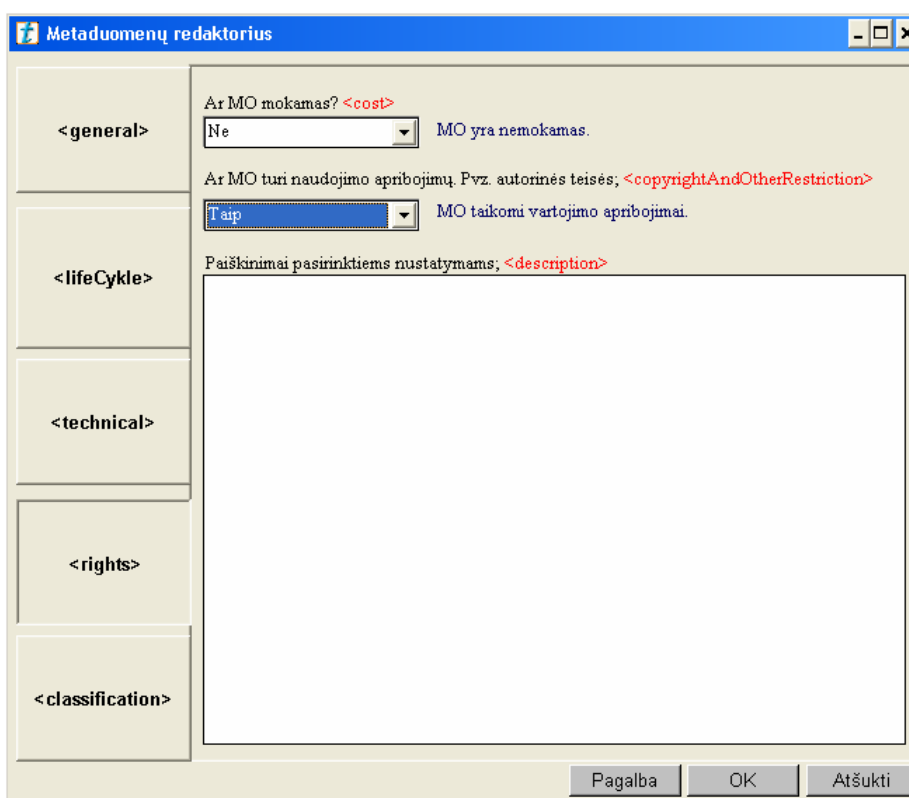
4.11 pav. <contribution> įrašo pridėjimas

- <technical> - vartotojui jokie nustatymai neprieinami, tiesiog jam pateikiama bendra infor-  
macija apie atliekamus veiksmus.



4.12 pav. Techninių <technical> duomenų aprašymas

- <rights> - vartotojas nurodo mokymo objekto apribojimus.



4.13 pav. Mokymo objekto apribojimų <rights> aprašymas

- <classification> - aprašoma pagal kokia sistema klasifikuojamas mokymo objektas

4.14 pav. Klasifikatoriaus <classification> aprašymas

	Identifikatorius <id>	<entry>
1		
2		
3		
4		
5		

4.15 pav. Klasifikatoriaus įrašų redagavimas

### 4.3. PROGRAMOS VEIKIMAS

Programos veikimas testuotas Windows 2000/XP sistemose. Teoriškai taip pat turėtų veikti ir kitose Windows platformose.

Programos paleidimui būtinas **Neuron** OpenScrip interpretatorius. Šis interpretatorius yra visiems laisvai prieinamas per internetą. Instaliacinis failas pateiktas kartu su programos failais.

Programos paleidimas:

- Jei nėra instaliuotas Neuron įskiepis (tiesiog paleidžiame instaliacinį failą Neuron.exe ir nurodome instaliacijos vietą);
- Paleidžiame programos paleidžiamą failą **MO\_pakavimas.tbk**.

#### 4.4. PROGRAMOS DUODAMAS REZULTATAS

Programa, atlikusi darbą, sukuria pilnavertę mokymo objektų SCORM kompoziciją. Sėkmingai atlikus darbą, paketą turėtų sudaryti šis failų rinkinys:

- Nurodyto mokymo objekto failas, pvz., 1 pamoka.html;
- Su įtrauktu mokymo objektu susiję failai, pvz., schema.jpg;
- adlcp\_rootv1p2.xsd;
- ims\_xml.xsd;
- imscp\_rootv1p1p2.xsd;
- imsmanifest.xml;
- imsmd\_rootv1p2p1.xsd;
- metadata.xml;
- MO.htm;
- SCORMLogika.js.

Visi šie failai yra surenkami į vieną katalogą, kuris pavadinamas taip:

[Mokymo Objekto vardas be priesagos]\_SCORM\_Paketas

Paskutiniame programos žingsnyje vartotojas sukurtą SCORM kompoziciją gali suarchyvuoti ZIP formatu (SCORM standarte tai specifikuojama kaip PIF failas). Suarchyvuotas failas bus patalpintas toje pačioje direktorijoje, kaip ir mokymo objektas, ir bus pavadintas:

[Mokymo Objekto vardas be priesagos]\_PIF.zip



## 5. IŠVADOS

1. Darbe atlikta analizė parodė, kad pasaulinėje rinkoje elektroninio mokymo sfera sparčiai plečiasi ir auga, todėl tampa aktualu sukurti elektroninių kursų mokymo objektų komponavimo sistemą, nepriklausomą nuo pačios mokymo medžiagą pateikiančios sistemos. Preke tampa ne tik pats kursas, bet ir mokymo objektai.
2. Tikslas įgyvendinimas remiasi SCORM standartu, nurodančiu, kaip mokymo medžiaga turi būti pateikiama ir sukuriamą. Tai leidžia užtikrinti kurso ilgaamžiškumą, panaudojamumą, keičiamumą, pritaikomumą.
3. Kadangi šio standarto kursai aprašomi XML, dar vienas sukurto įrankio privalumas – vartotojo eliminavimas iš tiesioginio XML redagavimo.
4. Pasirinktas paketinis sistemos projektavimas lengvai leidžia sistemą praplėsti, nekeičiant dabartinės struktūros, pridėdant agregavimo komponentus ar net elektroninio kurso testavimo modulis. Taip pat pereinant prie naujesnės SCORM standarto versijos.
5. Tolesniam sistemos tobulinimui numatyta mokymo objektų organizavimo tvarkos keitimas vartotojo lygmenyje (agregavimo modulio pridėjimas).
6. Buvo atlikti šie darbai:
  - a) analitinis darbas identifikuojant problemą;
  - b) detalizuotas ir schematizuotas komponavimo procesas;
  - c) sudarytas sistemos paketinis projektas;
  - d) sukurtas vartotojo sąsajos modelis;
  - e) realizuotas elektroninių kursų mokymo objektų komponavimo įrankis;
  - f) eksperimentinis sukurtos sistemos tyrimas;
  - g) eksperimentinis sistemos duodamų rezultatų teisingumo tyrimas.
7. Eksperimento rezultatai leidžia daryti išvadą, jog:
  - a) XML struktūros atitinka SCORM 1.2 reikalavimus;
  - b) kurso pakavimas atitinka visus SCORM 1.2 reikalavimus;
  - c) mokymo objekto aprašymas visiškai atskirtas nuo konkrečios mokymo valdymo sistemos;
  - d) SCORM standartas leidžia funkcionaliai ir aiškiai aprašyti mokymo objektus.
8. Naudojant sistemą realaus mokymo objektų komponavimui, galima bus, pridėti daugiau pasirinkimo funkcijų, vartotojui leisti pačiam organizuoti mokymo objektus, kurti sudėtingas struktūras. Tam būtinas agregavimo modulis.
9. Darbo tikslas įgyvendintas, nes elektroninių kursų mokymo objektų komponavimo sistema atitinka keltus reikalavimus ir pakankamai gerai atlieka mokymo objektų komponavimą.

## 6. LITERATŪRA

1. Aušra A. Mokslinė elektroninė biblioteka kaip e-studijų priemonė. [žiūrėta 2005-08-10]. Prieiga per internetą: [http://www.elibrary.lt/ktu\\_konf\\_pranes.phtml](http://www.elibrary.lt/ktu_konf_pranes.phtml);
2. Distancinio mokymo metodiniai tyrimai. [žiūrėta 2003-01-04]. Prieiga per internetą <http://metdm.ktu.lt>.
3. Elektroninio mokymosi medžiagos ruošimas. [žiūrėta 2004-03-09]. Prieiga per internetą: <http://distance.nsc.vu.lt>.
4. Gudas S. Organizacijų veiklos modeliavimas: vadovėlis. Kaunas, 2002. p.126.
5. Kovertaitė V.R. Mokomosios medžiagos rengimas. [žiūrėta 2004-03-03]. Prieiga per internetą: [http://www.liedm.lt/about/metodiniai\\_nurodymai/modulis2\\_index.html](http://www.liedm.lt/about/metodiniai_nurodymai/modulis2_index.html).
6. Ostyn C., Cooking up a SCORM: A SCORM 1.2 Content Cookbook for Developers. 2001 Click2Learn;
7. Packaging and publishing learning objects: best practice guidelines. 2005 Becta;
8. Programming in OpenScript. 2004 Sum Total.
9. Rutkauskienė D. Įvadas į distancines studijas: paskaita. [žiūrėta 2003-04-10]. Prieiga per internetą: <http://distance.ktu.lt/vips/sesionrecords.php>.
10. Rutkauskienė D. Nuotolinis mokymasis (Abarius P., Mačiulis M. Virtuali mokymosi terpė WebCT). Kaunas: Technologija, 2003.
11. Rutkauskienė D. Nuotolinis mokymasis (Kovertaitė V.R. Mokomosios medžiagos rengimas.). Kaunas: Technologija, 2003.
12. Rutkauskienė D. Nuotolinis mokymasis (Kubiliūnas R., Cibulskis G. E. mokymosi kurso aplinka CDK)). Kaunas: Technologija, 2003.
13. SCORM® 2004 2nd Edition Overview. ADL Technology Center;
14. SCORM® Run-Time Environment (RTE) Version 1.3.1. ADL Technology Center;
15. SCORM® Version 1.2 Content Aggregation Model. ADL Technology Center. 2003.
16. SCORM® Version 1.3.1 Content Aggregation Model. ADL Technology Center. 2004;
17. Šakys V. Informacinės technologijos: vadovėlis XI-XII klasei. I ir II knyga. Kaunas: Šviesa, 2004.
18. Targamadžė A., Normatas E., Rutkauskienė D., Vidžiūnas A. Naujos distancinio švietimo galimybės. – Vilnius, 1999.
19. Valstybės Žinios: Lietuvos nacionalinė informacinės visuomenės plėtros koncepcija. Vilnius, 2001, Nr. 20-652.

## 7. SANTRUMPŲ IR TERMINŲ ŽODYNAS

1. **Distancinis mokymas** – mokymas, kai yra bendraujama per atstumą: paštu, televizija, internetas.
2. **Elektroninė knyga** – elektroninis knygos variantas, dažniausiai pateikiamas pdf formatu.
3. **Elektroninė mokykla** – įprastinės mokyklos funkcijos perkeltos į elektroninę erdvę.
4. **Elektroninis kursas** – įprastinis kursas, perkeltas į elektroninę erdvę, kurą pateikia MVS.
5. **Elektroninis mokymas** – mokymasis naudojant elektronines priemones, elektronines knygas, ryšį palaikant internetu.
6. **Grižtamasis ryšys** – ryšis su besimokančiais, reagavimas į besimokančiųjų veiklą ir pas-tangas. Grižtamasis ryšys gali būti palaikomas *apklausomis, paklausimais elektroniniu paštu ar diskusijų forumais*.
7. **Komponavimas** – elektroninio kurso aprašymas ir supakavimas pagal tam tikrą standartą.
8. **Manifestas** – elektroninio kurso struktūros, metaduomenų, eiliškumo ir mokymo objektų ap-rašas.
9. **Meta-duomenys** – duomenys, aprašantys mokymo objektus, kursus.
10. **Mokymo objektas** (asset) – mokymo resursai, kuriuos apjungiant ir organizuojant yra kuria-mas elektroninio mokymo kursas.
11. **MTVS** – mokymo turinio valdymo sistema.
12. **MVS** – mokymo valdymo sistema.
13. **Nuotolinis mokymas** – žiūrėti *distancinis mokymas*;
14. **PIF** (Package Interchange File) – tai suspaustas ZIP formato PKZip v2.04g archyvas, atitin-kantis RFC 1951 reikalavimus.
15. **SCORM** (Sharable Content Object Reference Model) – tai turinio panaudojimo objektinių modelių standartas, kuris apibrėžia Web technologijomis paremtus elektroninio mokymo ob-jektus, jų atvaizdavimą ir organizavimą.
16. **Turinio pakavimas** – visi reikalingi kurso ar mokymo objektų pateikimui failai, surenkami ir pateikiami kaip PIF failas.
17. **TVS** – turinio valdymo sistema.
18. **XML** (Extensible Markup Language) -
19. **ZIP** - bylos formatas. ZIP byla gali turėti vieną ar kelis failus, kurie yra suspausti ir saugomi.

## 8. PRIEDAI

### 8.1. PROGRAMOS IŠEITIES TEKSTO FRAGMENTAI

```
--Nurodyto mokymo objekto apdorojimas
to handle MO_Radimas
-- Aprasome sisteminius kintamuosius --
  system s_RastiMO, \
    s_MOPavadinmas, \
    s_KurtiDIR, \
    s_xmlMD
  system s_MOKelias
  sysCursor = 4
-----

  send UpdateStatusBar
  if s_RastiMO is null
    s_RastiMO = UserPreferenceLoad("RastiMOpavadinima")
  end
  if s_RastiMO is null
    s_RastiMO = UserPreferenceLoad("RastiMOKelia")
  end
  if charcount(s_MOPavadinmas) > 0
    fnm = s_MOPavadinmas
  else
    fnm = s_RastiMO
  end
end
-----

  local MO_Filtras[][]
  local WORD i
--Nurodome keleta pasirinkimu(filtra), kad lengviau butu rasti norima faila
-----

  increment i
  MO_Filtras[i][1] = ASYM_GetString("Visos bylos");      MO_Filtras[i][2] = "*"
  increment i
  MO_Filtras[i][1] = ASYM_GetString("Acrobat bylos"); MO_Filtras[i][2] = "pdf"
  increment i
  MO_Filtras[i][1] = ASYM_GetString("PowerPoint bylos");MO_Filtras[i][2]= "ppt"
  increment i
  MO_Filtras[i][1] = ASYM_GetString("HTML bylos");    MO_Filtras[i][2] = "htm"
-----

--Mokymo objekto isrinkimas
-----

  local stack filtras
  step i from item 1 of dimensions(MO_Filtras) to 1 by -1
    push "*" & MO_Filtras[i][2] onto filtras
    push MO_Filtras[i][1] & " (*. " & MO_Filtras[i][2] & ")" onto filtras
  end
  sysCursor = 1
  get openFileDialog32(ASYM_GetString("Nurodykite vieta, kurioje yra mokymo objek-
tas"), \
    ASYM_ShortFileName(fnm), pathOfFile(fnm), filtras, 1)
  if it is null
    break
  end
end
--Netinkamu failu tipu atmetimas
-----

  ext = GetFileNameExtension(it)
  if itemOffset(ext, "zip,exe,sit,js,com") > 0
    request ASYM_GetString("Deja negalima itraukti šito failo")
  end
end
```

```

        break
    end
    sysCursor = 4
    send resetAllFiles
--Issaugom reikalingus MO parametrus
-----
    s_RastiMO = it
    MO_par = UserPreferenceSave("RastiMOpavadinima", it)
    MO_par = UserPreferenceSave("RastiMOkelia", pathOfFile(it))

--Sukuriam kataloga ir perkialiam mokymo objekta
-----
    s_MOPavadinmas = it
    s_KurtiDIR = ASYM_PathOfFile(s_MOPavadinmas)\
        & fileNameMinusExtension(ASYM_ShortFileName(s_MOPavadinmas)) &
    "_SCORM_Paketas"
    send ClearStatusField
--Isvedam info apie pasirinkta MO
-----
    txt = ASYM_GetString("Nurodytas mokymo objekto failas:") & CRLF \
        & s_MOPavadinmas
    s_MOkelias = ASYM_PathOfFile(s_MOPavadinmas)
    send FilesAddFileToCopyFromSrcDir ASYM_ShortFileName(s_MOPavadinmas)

--Itraukinat HTML failus ikeliami ir susije katalogai jei yra
-----
    ext = GetFileNameExtension(s_MOPavadinmas)
    if ext is "htm" or ext is "html"
        send Prideti_Suijusius_Katalogus(s_MOPavadinmas)
    end

    get DisplayStatusText(txt) of field "FailoVieta" of this page
    if not ValidateBuildDirectory(s_KurtiDIR)
        sysCursor = 1
        break
    end
    get InitDefaultMetadata(s_MOPavadinmas)
    get button "Toliau" of this background
    enabled of it = true
    excludeTab of it = false
    default of it = true
    focus = it
    send UpdateStatusBar
    txt = txt & CRLF \
        & ASYM_GetString("Paketas bus sufomuotas šiame kataloge:")\
        & CRLF & s_KurtiDIR
    get DisplayStatusText(txt) of field "FailoVieta" of this page
    sysCursor = 1
end

--I paketa itraukiami susije su MO katalogai
-----
to handle Prideti_Suijusius_Katalogus fn
    system s_susijusieMOfaila
    system s_kopijuotiSusijusiusFailus[]

    local subdir
    p = revOffset(".",fn)
    if p > 1
        if ASYM_IsDirectory(chars 1 to p-1 of fn & "_bylos")
            MO_Katlogas = chars 1 to p-1 of fn & "_bylos"
        else
            fn = chars 1 to p-1 of fn & ".*"
            path = ASYM_PathOfFile(fn)
        end
    end
end

```

```

        lst = GetFileList32(fn)
        step i from 1 to textLineCount(fn)
            if ASYM_IsDirectory(path & textLine i of fn)
                MO_Katlogas = path & textLine i of fn
            end
        end
    end
end
end
if charCount(MO_Katlogas) > 1 and last char of MO_Katlogas is not "\"
    put "\" after MO_Katlogas
end
s_susijusieMOfaila = MO_Katlogas
if MO_Katlogas is not null
    lst = GetFileList32(MO_Katlogas & "**")
    step i from textLineCount(lst) to 1 by -1
        if char 1 of textLine i of lst is "."
            clear textLine i of lst
        end
    end
    fill s_kopijuotiSusijusiusFailus with lst in [textline] order
end
end

```

## 8.2. EKSPERIMENTINIO ELEKTRONINIO KURSO XML STRUKTŪRA

```

<?xml version="1.0" ?>
= <manifest identifier="1_pamoka.htm.200512301642-H3B91872-03CC-ED10-9816-
  3120055D76AD" version="20060102121130"
  xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2
    imscp_rootv1p1p2.xsd http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
    imsmd_rootv1p2p1.xsd http://www.adlnet.org/xsd/adlcp_rootv1p2
    adlcp_rootv1p2.xsd">
- <!--
Sukurtas 2006.01.02 12:11:30
Elektroninių kursų mokymo objektų komponavimo sistemos.
-->
= <metadata>
<schema>ADL SCORM</schema>
<schemaversion>1.2</schemaversion>
= <lom xmlns="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemalocation="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
    imsmd_rootv1p2p1.xsd">
= <general>
= <title>
<langstring xml:lang="x-none">Eksperimentinis interneto temų kursas</langstring>
</title>
= <description>
<langstring xml:lang="x-none">Kursas parenktas remiantis XI, XII klasių IT bendruotju
  kursu</langstring>
</description>
<language>lt</language>
= <keyword>

```

```

<langstring xml:lang="x-none">Žiniatinklis</langstring>
  </keyword>
= <keyword>
  <langstring xml:lang="x-none">internetas</langstring>
    </keyword>
    </general>
= <lifecycle>
= <version>
  <langstring xml:lang="x-none">1.0</langstring>
    </version>
= <status>
= <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
= <value>
  <langstring xml:lang="x-none">Final</langstring>
    </value>
  </status>
= <contribute>
= <role>
= <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
= <value>
  <langstring xml:lang="x-none">Author</langstring>
    </value>
  </role>
= <centity>
  <vcard>BEGIN:vCard VERSION:3.0 FN:Ričardas ORG:KTU LABEL:Krėvė̃s 114
    TEL;TYPE=VOICE,MSG,WORK:121221 EMAIL;TYPE=INTERNET:Ricas@centras.lt
    END:vCard</vcard>
  </centity>
= <date>
  <datetime>2006-01-02</datetime>
  </date>
  </contribute>
  </lifecycle>
= <metametadata>
  <metadatascheme>ADL SCORM 1.2</metadatascheme>
  </metametadata>
= <technical>
  <format>application/msword</format>
  <format>image/jpeg</format>
  <format>image/png</format>
  <format>text/html</format>
  <size>792015</size>
  <location>.</location>
  </technical>
<educational />
= <rights>
= <cost>
= <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
    </source>
= <value>

```

```

<langstring xml:lang="x-none">no</langstring>
  </value>
</cost>
= <copyrightandotherrestrictions>
= <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
  </source>
= <value>
  <langstring xml:lang="x-none">yes</langstring>
  </value>
</copyrightandotherrestrictions>
= <description>
  <langstring xml:lang="x-none">Naudojant būtina nuoroda į autorių</langstring>
  </description>
</rights>
= <classification>
= <purpose>
= <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
  </source>
= <value>
  <langstring xml:lang="x-none">Discipline</langstring>
  </value>
</purpose>
= <description>
  <langstring xml:lang="x-none">Kursas skirtas interneto temų įsisavinimui</langstring>
  </description>
= <keyword>
  <langstring xml:lang="x-none">Žiniatinklis</langstring>
  </keyword>
= <keyword>
  <langstring xml:lang="x-none">internetas</langstring>
  </keyword>
= <taxonpath>
  <source>Pirma pamoka</source>
= <taxon>
  <id>ID1</id>
= <entry>
  <langstring xml:lang="x-none">1 pamoka</langstring>
  </entry>
</taxon>
</taxonpath>
</classification>
= <classification>
= <purpose>
= <source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
  </source>
= <value>
  <langstring xml:lang="x-none">Idea</langstring>
  </value>
</purpose>
= <description>
  <langstring xml:lang="x-none">Kursas parenktas remiantis XI, XII klasių IT bendruotju
  kursu</langstring>

```



```

</description>
= <keyword>
<langstring xml:lang="x-none">Žiniatinklis</langstring>
</keyword>
= <keyword>
<langstring xml:lang="x-none">internetas</langstring>
</keyword>
</classification>
</lom>
</metadata>
= <organizations default="ORG_1">
= <organization identifier="ORG_1" structure="hierarchical">
<title>Content</title>
= <item identifier="ITEM1" identifierref="RES1" isvisible="true">
<title>Content</title>
</item>
</organization>
</organizations>
= <resources>
= <resource identifier="RES1" type="webcontent" adlcp:scormtype="sco" href="MO.htm">
<file href="MO.htm" />
<file href="tuscia.htm" />
<file href="SCORMLogika.js" />
<file href="1_pamoka.htm" />
<file href="1_pamoka_bylos/1.htm" />
<file href="1_pamoka_bylos/102%20uzd.doc" />
<file href="1_pamoka_bylos/2.htm" />
<file href="1_pamoka_bylos/3.htm" />
<file href="1_pamoka_bylos/4.htm" />
<file href="1_pamoka_bylos/image101.png" />
<file href="1_pamoka_bylos/image102.jpg" />
<file href="1_pamoka_bylos/image201.png" />
<file href="1_pamoka_bylos/image202.jpg" />
</resource>
</resources>
</manifest>

```

### 8.3. NAUDOJAMŲ XML APRAŠŲ PAIŠKINIMAS

XML aprašas	XML duomenų tipas	Paaiškinimas
<manifest>	Konteineris	Pagrindinis manifesto failo elementas
<metadata>	Konteineris	Metaduomenys apie manifesto failą. Gali būti iškelti iš imsmaniest.xml failo į metadata.xml
<schema>	String	Apibrėžia, pagal kokią schemą bus valdomas manifesto failas
<schemaversion>	String	Nurodo, kokia schemas versija naudojama
<lom>	Konteineris	Apima visus manifesto metaduomenis
<general>	Konteineris	Elementų grupė, kuri apibrėžia elementus kaip visumą
<title>	LangString	Mokymo objekto pavadinimas, antraštė
<langstring>		Apibrėžiami duomenys priklausomai nuo konteinerio.
<description>	LangString	Mokymo objekto žodinis apibrėžimas

<language>	CharacterString	Apibrėžia naudojamą kalbą
<keyword>	LangString	Raktiniai žodžiai, nusakantys mokymo objektą
<lifecycle>	Konteineris	Elementų grupė, nusakanti mokymo objekto būsenas
<version>	LangString	Nurodo mokymo objekto versiją.
<status>	Vocabulary	Nurodo mokymo objekto būseną.
<value>	String	Apibrėžiama vertė priklausomai nuo konteinerio
<contribute>	Konteineris	Apibrėžiama mokymo objekto esmė, paskirtis
<role>	Vocabulary	Nurodoma mokymo objekto paskirtis
<source>	Vocabulary	Nurodomas pasirinkimų pateikimo šaltinis
<entity>	CharacterString	Nurodomi mokymo objektą kuriantys, apdorojantys asmenys ar organizacijos
<date>	Konteineris	Nurodo <contribution> aprašymo datą ir aprašą
<datetime>	DataType	Nurodomas laikas
<technical>	Konteineris	Techniniai SCORM reikalavimai
<format>	CharacterString	Techninis duomenų tipas
<location>	CharacterString	Nurodo SCORM metaduomenų vietą
<educational>	CharacterString	Apibrėžia SCORM kurso pedagogines savybes
<rights>	Konteineris	Mokymo objekto apribojimai
<cost>	Vocabulary	Ar mokymo objektas mokamas
<source>	Vocabulary	Nurodomas pasirinkimų pateikimo šaltinis
<copyrightandotherrestrictions>		Mokymo objekto autorinės teisės ir kiti apribojimai.
<classification>	Konteineris	Mokymo objekto klasifikatorius
<purpose>	Vocabulary	Klasifikatoriaus parinkimo priežastis
<source>	Vocabulary	Nurodomas pasirinkimų pateikimo šaltinis
<taxonpath>	Konteineris	Apibrėžia klasifikavimo sistemą
<taxon>	Konteineris	Klasifikatoriaus elementas
<id>	CharacterString	Klasifikavimo elemento identifikavimo numeris
<entry>	LangString	<taxon> vardas, etiketė
<organizations>	Konteineris	Apibrėžia mokymo objekto organizavimo struktūrą
<resources>	Konteineris	Apibrėžia naudojamus resursus
<file href>	String	Nurodoma reali resurso vieta

#### 8.4. PRIDEDAMO CD TURINYS

1. Programos kūrimo failai;
2. Programos palaikymo programa Neuron.exe;
3. Elektroninis magistrinio tezių variantas;
4. Testavimo pavyzdys;
5. Naudotos elektroninės knygos.

### 8.5. E-KURSO SAŲVEIKŲ TARP KLIENTO IR SERVERIO SCHEMAS

