

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
KOMPIUTERINIŲ TINKLŲ KATEDRA**

**Daiva Kalvaitienė**

**Kompiuterinės programavimo įgūdžių testavimo  
sistemos tyrimas bei projektavimas**

Magistro darbas

**Vadovas  
doc. K. Baniulis**

**KAUNAS, 2005**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
KOMPIUTERINIŲ TINKLŲ KATEDRA**

**TVIRTINU  
Katedros vedėjas  
doc. dr. R. Plėštys  
2005-01-11**

**Kompiuterinės programavimo įgūdžių testavimo  
sistemos tyrimas bei projektavimas**

**Informatikos magistro baigiamasis darbas**

**Vadovas  
doc. K. Baniulis  
2005-01-14**

**Recenzentas  
doc. dr. J. Adomavičius  
2005-01-10**

**Atliko  
IFN -2 gr. stud.  
D. Kalvaitienė  
2005-01-06**

**KAUNAS, 2005**

## KVALIFIKACINĖ KOMISIJA

**Pirmininkas:** Raimundas Stulpinas, UAB „Strauja“ generalinis direktorius

**Sekretorius:** Antanas Lenkevičius, docentas;

**Nariai:** Rimantas Butleris, docentas,  
Valentinas Kiauleikis, docentas,  
Jonas Kazimieras Matickas, docentas,  
Bronius Paradauskas, docentas,  
Dalius Rubliauskas, docentas,  
Aleksandras Targamadžė, profesorius

## **Summary**

While organizing education at Lithuania and other countries universities, the IT are widely used at colleges. Indocinating these technologies one of the most complicated problem is automatical knowledge theoretical and practical evaluation of those who are studying.

In this work it is investigated the architecture of the testing system of programming skills. Here also are discussed the main characteristic and applying prospects of such system. The standards of electronical teaching are talked over as well as how they are adapted in the system for making questionnaires.

It is submitted how the problem of testing practical skills in the subjects of programming may ching (training) systems, and languages compilers of the programming.

In order to make the test of programming skills chose already present KTU (Kaunas Technologies University ) electronical teaching system. This my decision was accepted for such reason because already some years this system is used successfully and is built up by using modern programming technologies and has already created the model of standardization.

To my way of thinking more and more widen using distance training ways only with the help of such system will be possible grandly to evaluate theoretical and practical knowledge's of the students.

## TURINYS

<b>1. ĮŽANGA.....</b>	<b>4</b>
1.1. PRAKTIŅŲ ĮGŪDŲŲ TESTAVIMO SISTEMOS KŪRIMO AKTUALUMAS .....	4
1.2. PROGRAMAVIMO ĮGŪDŲŲ MOKYMO KONTROLĖS PROBLEMIŠKUMAS.....	5
<b>2. TIKSLAI IR UŖDAVINIAI.....</b>	<b>6</b>
2.1. TIKSLAS .....	6
2.2. UŖDAVINIAI .....	6
2.3. TAIKOMOJI SRITIS .....	6
<b>3. KOMPIUTERINIŲ TESTAVIMO SISTEMŲ ANALIZĖ .....</b>	<b>7</b>
3.1. TESTAVIMO BŪDAI.....	7
3.2. PASAULYJE NAUDOJAMŲ ELEKTRONINIŲ MOKYMO SISTEMŲ APŖVALGA.....	8
3.2.1. <i>Testavimo sistemos skirtos programavimo varžyboms, olimpiadoms</i> .....	9
3.2.2. <i>Programavimo varžybų aptarnavimo sistemos (PVAS)</i> .....	10
3.2.2.1. PVAS naudojimas Lietuvoje .....	11
3.2.3. <i>Virtualios mokymosi aplinkos</i> .....	12
3.2.4. <i>EMS apžvalgos rezultatai</i> .....	14
3.3. ELEKTRONINIO MOKYMO STANDARTAI .....	16
3.3.1. <i>Metaduomenų standartai</i> .....	19
3.3.2. <i>Mokymo objektų standartai</i> .....	20
3.3.3. <i>Standartai mokymo medžiagos tikrinimui</i> .....	26
3.4. ELEKTRONINIO MOKYMO PERSPEKTYVA .....	27
<b>4. PROGRAMAVIMO ĮGŪDŲŲ TESTAVIMO SISTEMOS PROJEKTAVIMAS .....</b>	<b>28</b>
4.1. REIKALAVIMAI PROGRAMAVIMO ĮGŪDŲŲ TESTAVIMO SISTEMAI.....	28
4.1.1. <i>Reikalavimai programavimo įgūdžių kontrolės užduočių tipams</i> .....	28
4.2. KOMPIUTERIZUOTOS TESTAVIMO SISTEMOS STRUKTŪRA .....	29
4.3. TESTAVIMO SISTEMOS POSISTEMIŲ FUNKCINIAI MODELIAI .....	30
4.4. XML DUOMENŲ STRUKTŪRA KLAUSIMAMS IR TESTAMS APRAŠYTI .....	35
4.5. PROGRAMINĖS REALIZACIJOS PROJEKTAVIMAS .....	36
4.5.1. <i>Redaktoriaus - XML konverterio sudarymas</i> .....	37
4.5.2. <i>Kompiliatoriaus įdiegimas</i> .....	37
<b>5. PROGRAMAVIMO ĮGŪDŲŲ TESTAVIMO SISTEMOS EKSPERIMENTINIS TYRIMAS.....</b>	<b>39</b>
5.1. UŖDUOČIŲ TIPŲ REALIZAVIMAS.....	39
5.1.1. <i>Ekspimentinis redaktoriaus-XML konverterio panaudojimas</i> .....	42
5.2. ATSKIRŲ PROCEDŪRŲ BEI FUNKCIJŲ TESTAVIMAS .....	43
5.3. EKSPERIMENTINIS TESTŲ STANDARTIZAVIMAS IMS QTI SPECIFIKACIJA.....	44
<b>IŠVADOS .....</b>	<b>47</b>
<b>LITERATŪROS SĄRAŠAS .....</b>	<b>48</b>
<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS.....</b>	<b>49</b>
<b>PRIEDAS 1 STANDARTAS TEISINGO ATSAKYMŲ PASIRINKIMO KLAUSIMUI.....</b>	<b>50</b>

## LENTELIŲ SĄRAŠAS

3.1 lentelė: Testavimo sistemų palyginimas .....	15
3.2 lentelė: Mokymo objektų funkcijos realizuojančios būtinus kreipinius į API sąsają.....	20

## PAVEIKSLĖLIŲ SĄRAŠAS

3.1 pav. Testų tipai pagal pateikimo būdą .....	7
3.2 pav. IMS mokomosios medžiagos talpinimo į paketą modelis .....	22
3.3 pav. Manifesto elementai [11].....	24
4.1 pav. Bendras sistemos TestTool4 veikimo modelis .....	31
4.2 pav. Dėstytojo posistemės funkcionalumo schema .....	32
4.3 pav. Studentų posistemės funkcionalumo schema .....	33
4.4 pav. Administratoriaus posistemės funkcionalumo schema .....	33
4.6 pav. Dėstytojų posistemės modulio veikimo schema .....	34
4.7 pav. Redaktoriaus-XML konverterio veikimo modelis.....	37
4.8 pav. Testavimo struktūra, įkomponavus kompiliatorių .....	38
4.9 pav. Sistemoje naudojamos reliacinės duomenų bazės struktūra .....	38
5.1 pav. Vieno ar kelių atsakymų pasirinkimo testas.....	39
5.2 pav. Klausimų analizės testas.....	40
5.3 pav. Kompilatoriaus panaudojimas praktinei užduočiai atlikti.....	41
5.4 pav. Pranešimų apie klaidas tekstinis failas .....	41
5.5 pav. Duomenų redagavimo ir konvertavimo įrankis .....	42
5.6 pav. Konvertuotas programos tekstas autoriaus ir studento aplinkoje.....	43

## 1. IŽANGA

Organizuojant mokymą, Lietuvos ir kitų šalių universitetuose, kolegijose plačiai naudojamos informacinės technologijos. Diegiant šias technologijas, viena sudėtingiausių problemų yra automatinis besimokančiųjų praktinių žinių įvertinimas. Tikrinant mąstymo ir uždavinių sprendimo įgūdžius, svarbūs ne tik galutiniai rezultatai, bet ir sprendimo eiga, dalinio problemos sprendimo įvertinimas.

Šiame darbe nagrinėjama programavimo įgūdžių testavimo sistemos architektūra, aptariamoms pagrindinėms tokios sistemos savybėms bei taikymo perspektyvos; apžvelgiami elektroninio mokymo standartai ir jie pritaikomi sistemoje testų klausimų sudarymui. Pateikiama kaip praktinių įgūdžių testavimo problema programavimo disciplinose dalinai galima išspręsti įkomponuojant į elektroninio mokymo sistemas programavimo kalbų kompiliatorius.

Programavimo įgūdžių testavimui atlikti pasirinkau jau esamą KTU elektroninio mokymo sistemą. Šį mano sprendimą lėmė tai, kad ji jau keletą metų sėkmingai naudojama, yra sukurta panaudojant šiuolaikines programavimo technologijas, turi sudarytą standartizavimo modelį.

Manau, kad vis plačiau naudojant nuotolinio mokymo būdus, tik tokių sistemų pagalba bus galima kokybiškai įvertinti tiek teorines tiek praktines studentų žinias.

### 1.1. Praktinių įgūdžių testavimo sistemos kūrimo aktualumas

Studijų procese įgytų žinių vertinimas ir tikrinimas kompiuteriniu testavimu taikomas vis plačiau. Kompiuterinių testų pagalba galima sparčiai gauti reikalingą ir pakankamai patikimą informaciją apie testuojamąjį. Testas, kaip žinių vertinimo ir tikrinimo forma, leidžia išvengti subjektyvaus vertinimo būdingo kitiems žinių bei gebėjimų tikrinimo būdams. Kompiuteriniame teste su testuojamuoju bendrauja kompiuterinės programinės priemonės ir tai psichologiškai patrauklu bei priimtina didžiajai daugumai šiuolaikinių studentų.

Dėstant programavimo disciplinas, studentams pateikiamas didelis naujų sąvokų skaičius ir yra labai svarbūs jų įgytų žinių praktinio taikymo įgūdžiai, todėl aktuali praktinių programavimo įgūdžių kompiuterinio testavimo sistemų sukūrimo problema. Dalinai šią problemą galima išspręsti naudojant tradicinius apklausos testus, papildytus su pateikiamais klausimais susijusios situacijos aprašymais. Pateikiant testuose sudėtingesnių uždavinių sprendimų užduotis, tikslinga testuojamajam leisti pasinaudoti tipinėmis programų verifikavimo priemonėmis, kurios yra visose programavimo aplinkose. Tai nesunkiai galima

padaryti įkomponuojant į testavimo sistemas programavimo kompiliatorius. Be to, tokiose sistemose galima gerokai padidinti jų aptarnaujamų testų įvairovę, o tuo pačiu ir testavimo sistemos efektyvumą.

Kaip jau buvo pastebėta, testavimo sistemų taikomoji sritis yra labai plati. O privalumas, darantis tokio tipo sistemas populiariomis ir skatinantis jomis naudotis, yra užduočių paruošimo bei pateikimo supaprastinamo galimybė. Testuotojas (dėstytojas) užduočių pateikimui gali naudotis įrankiais kurie yra integruoti pačioje sistemoje. Kiek sudėtingiau yra vertinti rezultatus. Kompiuteris ne visas užduotis gali įvertinti teisingai, pav. tikrinant programos kodą, negalima apie pačios programos tikslumą spręsti vien iš klaidų buvimo arba nebuvimo. Tad šiuo atveju testavimo sistema gali pateikti tik preliminarius testų rezultatų įvertinimus, kurių tikslumas dažniausiai priklauso nuo pateiktos užduoties konkretumo.

## **1.2. Programavimo įgūdžių mokymo kontrolės problemiškas**

Nors pasaulyje egzistuoja daugybė programinės įrangos priemonių, leidžiančių kurti mokymo medžiagą, atlikti žinių testavimą, bet jos dažniausiai siūlo tik laikiną problemos sprendimą ir neužtikrina bendro mechanizmo, kuris teiktų efektyvią mokymo išteklių paiešką ir prieinamumą, daugkartinį turinio ir metodų panaudojimą, galimybę atnaujinti mokymo medžiagą, užtikrintų suderinamumą, saugumą ir panašiai. Tokių globalių problemų sprendimą siūlo Europos Sąjungos rekomenduojami standartai.

Dėstytojams reikia ne tik surinkti ir paruošti teorinę mokymo dalį, bet ir kurti klausimus elektroniniams testams. Tai užima daug laiko, bet problema gali būti išsprendžiama sukuriant duomenų saugyklas (dėstytojai galėtų naudoti ne tik savo sukurtus testus bei klausimus), kuriose saugomi mokymo objektai bei mokomoji medžiaga. Savaiame suprantama, kad visa tai sėkmingai veiktų, reikia, jog visa mokomoji medžiaga atitiktų tam tikrus standartus. Kadangi visa tai bus pateikiama kažkokios elektroninio mokymo sistemos pagalba, būtina jog ir pati mokymo sistema atitiktų pasaulinius standartus. Ir antras būdas padedantis išspręsti šią problemą – tai lanksčių testų sudarymo įrankių kūrimas, leidžiančių sudaryti įvairių tipų užduotis, atitinkančias specialią struktūrą.

Universalios testavimo sistemos yra įkomponuotos daugumoje nuotoliniams mokymui skirtų programinės įrangos komplektų, laisvai platinamos Internetu, tačiau jos gerai tinka tiktai faktinių žinių kontrolei ir stokoja loginio mąstymo ir gebėjimo spręsti įvairius uždavinius kontrolės priemonių. Todėl praktinio programavimo įgūdžių, kuriuos lemia gebėjimai logiškai mąstyti ir pritaikyti įgytas žinias sprendžiant įvairius uždavinius, kontrolei universalių testavimo sistemų panaudojimo galimybės yra labai ribotos.



## **2. TIKSLAI IR UŽDAVINIAI**

### **2.1. Tikslas**

Pateikti modelį, kaip pritaikyti elektroninio mokymo sistemą programavimo įgūdžiams testuoti.

### **2.2. Uždaviniai**

- Išanalizuoti analogiškų sistemų veikimo principus bei svarbiausias funkcijas;
- sudaryti programavimo įgūdžių testavimo sistemos konstravimo principus;
- sukurti vartotojui patogias priemones testų klausimams sudaryti bei konvertuoti į XML dokumentą.
- sukurti priemones praktiniams programavimo įgūdžiams testuoti.
- išanalizuoti elektroninio mokymo sistemų testavimo standartus bei juos pritaikyti programavimo testų klausimų sudarymui.

### **2.3. Taikomoji sritis**

Testavimas – viena lanksčiausių žinių tikrinimo formų. Kaip ir reikėtų tikėtis, tokios sistemos turi didelį pasisekimą švietimo įstaigose, kuriose besimokančiųjų žinių testavimas yra neatsiejama jų veiklos dalis.

Kompiuterizuota programavimo įgūdžių testavimo sistema bus naudojama KTU studentų žinių patikrinimui.

Žinių tikrinimas testais gali būti ir visiškai įdomi laisvalaikio praleidimo forma.

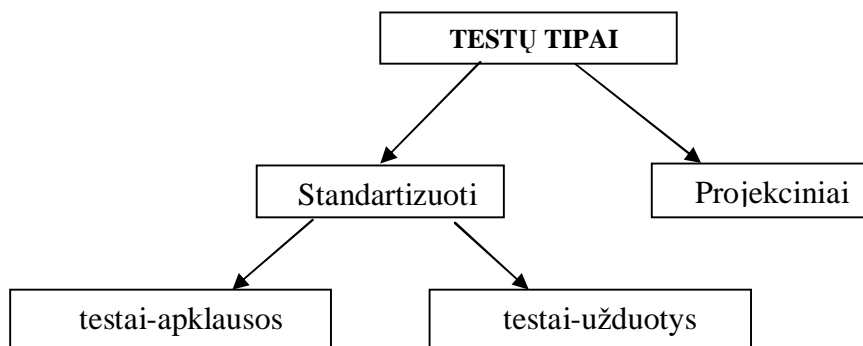
### 3. KOMPIUTERINIŲ TESTAVIMO SISTEMŲ ANALIZĖ

Siekiant užsibrėžtų tikslų realizavimo, analizuojami testavimo būdai, analogiškų testavimo sistemų darbo principai bei funkcinės galimybės ir su jomis susijusios problemos.

#### 3.1. Testavimo būdai

Testavimas laikomas vienu esminių psichologinės diagnostikos metodų, suteikiančiu galimybę gauti testuojamojo tobulėjimo tam tikroje srityje kokybinius bei kiekybinius rodiklius. Testai (nuo angl. test - bandymas) – tai standartizuotos psichodiagnostikos metodikos, paremtos santykinai trumpų išbandymų (užduočių, klausimų, situacijų ir pan.) testuojamajam pateikimu. Tokių išmėginimų pagalba su tam tikra tikimybe galime vertinti testuojamojo psichologines savybes, reikalingų teorinių žinių lygį, jo praktinius sugebėjimus bei kitas charakteristikas. Testų rezultatai leidžia vertinti testuojamuosius visos grupės mastu. Pats metodas yra patogus tuo, jog leidžia surinkti pirminius duomenis, juos apdoroti, atlikti kokybinę analizę bei interpretuoti jos rezultatus. Šiuo atveju “pirminiais duomenimis” laikomi duomenys, gauti testavimo metu ir neapdoroti matematiniais metodais.

Pagal pateikimo formą testai skirstomi į du pagrindinius tipus: standartizuotus ir projekcinius. Standartizuoti savo ruožtu dar skirstomi į du potipius: testus-apklausas ir testus-užduotis (žr. pav 3.1).



3.1 pav. Testų tipai pagal pateikimo būdą

Projekciniai testai atliekami su tikslu testuojamojo elgesio savybėms nagrinėti, todėl žinių tikrinimui netinka.

Kur kas priimtinesnis yra standartizuotų testų modelis. Šiam tipui priklauso testai, praėję standartizacijos etapą (yra patikrintas jų tikslumas, naudingumas bei kiti testams keliami reikalavimai). Tokie testas būna orientuoti į įvertinimą, todėl gali būti taikomi žinių tikrinimui. Standartizuotų testų pranašumas yra galimybė juos taikyti skirtingoms testuojamųjų grupėms, galima sudaryti testus, atitinkančius tam tikrą tikrinamų žinių lygį bei pobūdį. Standartizuotų testų pagalba

gaunami kokybiniai testuojamojo žinių įvertinimai, pagal kuriuos galima spręsti apie jo teorinį bei praktinį pasirengimą.

Deja, aptartas testavimo metodas neapsieina be trūkumų. Vienas iš jų – tai galimybė testuojamajam įtakoti testų rezultatus. Tai pasireiškia tuo atveju, kai testuojamasis žino rezultatų vertinimo metodiką, o kartais ir testo turinį.

Kitas standartizuotų testų trūkumas yra sąlygojamas testavimo proceso. Stresui labiau pasiduodantys žmonės dažniau praranda savikontrolę, jie pradeda jaudintis, klysti atsakinėdami į paprasčiausius klausimus. Tuo galima įsitikinti tarp lengvesnių užduočių įterpiant vieną pakankamai sunkią ir stebint testuojamųjų reakciją.

Standartizuotų testų sąvoka yra gana plati. Ji gerai tinka testavimui kaip procesui aprašyti, parodyti, kuo toks metodas yra naudingas, kokie jo pranašumai ir trūkumai, lyginant su kitais tyrimų būdais. Reikia pastebėti, kad skirtingų testų tyrimo objektai taip pat yra skirtingi, todėl toliau aptarsiu testų grupę, kuri orientuota į testuojamojo žinių, įgytų mokymo procese, įsisavinimo tyrimą. Tokie testai paprasčiausiai ir vadinami žinių įsisavinimo testais. Šio tipo testai kuriami mokomosios medžiagos pagrindu ir yra skirti vertinti testuojamojo žinių įsisavinimo lygį, praktinius įgūdžius, profesinį pasirengimą. Žinių įsisavinimo testai taikomi kolektyvui (klasei, kursui). Daugelyje mokymo įstaigų testai keičia egzaminavimo žodžiu sistemą.

Visiems, tame tarpe ir žinių įsisavinimo testams, yra keliami atitinkami reikalavimai. Vienas pagrindinių – testo tikslumas. Tai vienas iš pagrindinių testo kokybės kriterijų, parodančių, kiek testas yra tinkamas atlikti tyrimams nagrinėjamoje srityje. Testo tikslumas yra nustatomas tyrinėjant jo rezultatų koreliaciją. Pavyzdžiui, atlikus grupės testavimą naudojant naują testą, gauti rezultatai lyginami su tos pačios grupės testavimo rezultatais, gautais taikant jau laiko patikrintą testavimo metodiką. Teigiama, jog sudarytas testas yra pakankamai tikslus, kai koreliacijos koeficientas yra 0.6. Koeficiento reikšmėms kintant [0.45;0.65] testo tikslumas laikomas pilnai pakankamu.

Kitas svarbus reikalavimas testui – jo patikimumas. Kuo testas patikimesnis, tuo jis mažiau priklauso nuo matavimo paklaidų. Testo patikimumas apibūdinamas rezultatų stabilumu, pakartotinai testuojant tas pačias testuojamųjų grupes. Pakartotinis testavimas paprastai būna atliekamas praėjus 2 savaitėms nuo pirmojo. Rezultatų koreliacijos koeficiento reikšmei esant 0.75 teigiama, kad testo patikimumas pilnai pakankamas.

### **3.2. Pasulyje naudojamų elektroninių mokymo sistemų apžvalga**

### 3.2.1. Testavimo sistemos skirtos programavimo varžyboms, olimpiadoms

Pirmasis testavimo automatizavimas būdavo .bat bylos. Jos paleisdavo testuojamą programą, jai pateikdavo reikiamas pradinių duomenų bylas, išsaugodavo ir išanalizuodavo programos pateiktus rezultatus. Jei testuojama programa viršydavo vykdymo laiką, vykdymą tekdavo nutraukti pačiam vertintojui. Vartotojo sąsaja buvo labai primityvi. Dažnai galutiniai rezultatai būdavo rašomi ant popieriaus. Toks testavimo būdas iki šiol dažnai naudojamas mokyklose, labai nedideliuose konkursuose.

Antroji pakopa – testavimo sistemų kūrimas. Šios sistemos kartais būdavo tinklinės (t.y. dalyvių darbai būdavo testuojami dalyvių kompiuteriuose arba per tinklą paimami į vertintojų kompiuterį). Vartotojo sąsaja būdavo patogesnė, lengviau ištestuoti didelį kiekį programų, pertestuoti atskiras programas. Šios sistemos nutraukdavo per ilgai dirbančios programos vykdymą.

Testavimo sistemų darbo principai ir svarbiausios atliekamos funkcijos [5]:

- Testavimo sistema paleidžiama pasibaigus varžyboms.
- Dalyvis pateikia į diskelį įrašytą programą (programos tekstą arba sukompiliuotą programą) arba jo sprendimas pasiekiamas per tinklą.
- Jei reikia, pateiktoji programa kompiliuojama.
- Programa vykdoma su visais pradinių duomenų rinkiniais.
- Įsimenami visų testų, kuriems programa sėkmingai baigė darbą, rezultatai.
- Tikrinamas įsimintų rezultatų teisingumas:
  - sulyginant teisingą ir gautą rezultatus, jei atsakymas vienareikšmis;
  - įvykdant programą, kuri analizuoja moksleivio pateiktus rezultatus.

Šios sistemos leido greitai testuoti varžybų dalyvių programas, tačiau kilo daug problemų, susijusių su procedūriniais reikalavimais, kurių būtina laikytis norint automatizuotai testuoti sprendimus. Šių reikalavimų nesilaikančias (nors ir iš esmės teisingas) programas testavimo sistemos vertindavo nulių balų.

Testavimas informatikos olimpiadose suprantamas kaip kelių (ar keliolikos) pradinių duomenų parinkimas, vertinamos programos įvykdymas su šiais duomenų rinkiniais ir programos pateiktų rezultatų analizavimas.

Vertinimo kokybė olimpiadų darbus testuojant ir juos vertinant ne automatiniu būdu (analizuojant) skiriasi. Daugeliu atveju analizuojant sprendimą vertinimas yra objektyvesnis, nei testuojant. Pavyzdžiui, algoritmas yra iš esmės būti geras, tačiau užrašant algoritmą programa padaroma nedidelė klaida ir testuojant už šią programą balų neskiriama arba jų skiriama nedaug. Kitas pavyzdys: algoritmas yra euristinis arba jame yra esminė klaida, tačiau programa sėkmingai įveikia visus testus ir gauna maksimalų įvertinimą. Gal būt tai reiškia, kad

nepakankamai gerai parengti testai. Testuojant negalima įvertinti nei programavimo stiliaus, t.y. duomenų struktūrų parinkimo, algoritmo efektyvumo ir kai kuriose situacijose laiko požiūriu, programos teksto išdėstymo.

Tarptautinėse informatikos olimpiadose dėl daugiakalbystės ir darbų gausumo darbai vertinami tik juos testuojant. Lietuvos informatikos olimpiadose, vertinamas ir algoritmo idėjos aprašymas bei programavimo stilius. Taigi, Lietuvos informatikos olimpiadų kontekste, darbo vertinimas yra platesnė sąvoka nei vien jo testavimas.

Pasaulinio lygio varžybose buvo pereita prie naujos kartos Programavimo varžybų aptarnavimo sistemų, kurios iš esmės išsprendė su procedūriniais reikalavimais susijusias problemas.

### **3.2.2. Programavimo varžybų aptarnavimo sistemos (PVAS)**

Testavimo sistemas pakeitė platesnes galimybes turinčios *Programavimo varžybų aptarnavimo sistemos*. [5] PVAS paleidžiama varžybų pradžioje ir veikia visą varžybų laiką. PVAS turi žiniatinklio sąsaja, būtinai yra tinklinė ir dalyviai ją pasiekia naudodamiesi vietiniu tinklu ar internetu. Kartu su užduotimis dalyviams pateikiami iš jiems iš anksto nežinomi vartotojo vardai ir slaptažodžiai. Dalyviai prisijungia prie sistemos ir ja gali naudotis visą varžybų laiką. Dalyviai savo programas pateikia tik per PVAS. Be sprendimų pateikimo galimybės, PVAS gali atlikti kitas varžybų metu reikalingas funkcijas: [5]

- sprendimų (programų) priėmimas informuojant apie procedūrinius pažeidimus;
- programų testavimas vertinimo serveryje su dalyvio parengtais testais; tai leidžia dalyviui geriau įvertinti jo programos efektyvumą;
- sprendimų bei darbinių bylų saugojimas serveryje (apsaugoma nuo situacijos, jei varžybų metu sugestų dalyvio kompiuteris);
- galimybė užduoti klausimus ir gauti atsakymus;
- programos teksto atspausdinimas popieriuje;
- varžybų medžiagos (dalyvio pateiktų sprendimų, testų, pavyzdinių sprendimų) parsisiųsdinimas pasibaigus varžyboms.

PVAS funkcijos skirtos varžybų organizatoriams: [5]

- varžybų dalyvių registravimas;
- varžybų užduočių bei kitos reikalingos informacijos (pvz. free paskalio žinyno) pateikimas patogiu elektroniniu pavidalu;
- varžybų situacijos stebėjimas (galima stebėti kiek ir kada, konkretus dalyvis pateikė programų, kiek kartų sėkmingai ar nesėkmingai bandė pateikti sprendimus);

- techninės situacijos stebėjimas ir valdymas;
- sistemos saugumo užtikrinimas;
- galimybė pasiųsti pranešimus visiems varžybų dalyviams (pvz.: patikslinti sąlygą ar priminti, kad baigiasi varžybų laikas);
- automatizuotas sprendimų surinkimas, užtikrinant, kad visi priimti sprendimai tenkina procedūrinius reikalavimus;
- testavimas (atlieka visas ankstesniame skyrelyje išvardintas testavimo funkcijas);
- rezultatų gavimas (parengiami kiekvieno dalyvio rezultatų protokolai, bendros lentelės, statistika);

Kai programavimo varžybas aptarnauja PVAS, prasidėjus varžyboms, dalyviai prisijungia prie PVAS. Jei dalyvis išsprendė uždavinį (visą uždavinį ar tik dalį, tačiau turi veikiančią programą), jis gali iš karto sprendimą pateikti PVAS. Gavusi sprendimą sistema atlieka šiuos veiksmus:

- patikrina antraštę, kurioje nurodytas uždavinio identifikatorius ir programavimo kalba;
- sukompiluoja programą;
- įvykdo sukompiliotą programą su sąlygoje esančiais vienu ar daugiau pavyzdinių testų ir įsimena rezultatus;
- patikrina ar rezultatai atitinka reikalaujamą formatą;
- patikrina, ar rezultatai teisingi;

Jei visi šie etapai įveikiami sėkmingai, darbas priimamas, jei kuris nors etapas neįveikiamas, PVAS nurodo kuriame etape tai nutiko ir pateikiamas pranešimas apie problemą. Sprendimas nepriimamas. Dalyvis gali sutvarkyti problemą ir vėl pateikti sprendimą sistemai (darbą sistemai gali pateikti bet kuriuo varžybų momentu ir neribotą skaičių kartų, vertinimas nuo to nepriklauso).

### **3.2.2.1. PVAS naudojimas Lietuvoje**

Yra sukurta ne viena PVAS (Amerikiečių, Estų, Kiniečių ir Korėjiečių kurtos sistemos). Lietuvoje naudojama Korėjoje HM Research Ltd. sukurta sistema „Automatic contest & grading system“. Ši sistema dirba Linux operacinėje sistemoje. Sistemą adaptuoti Lietuvos informatikos olimpiadų poreikiams leido licenzija.[13] Todėl buvo lokalizuota vartotojo sąsajos, įvestos dvi skirtingos dalyvių grupės, t. y. joms numatytos skirtingos užduotys. PVAS papildyta galimybe dalyvius registruoti internetu, suteikta galimybė varžybų metu stebėti kiek uždavinių ir kokie dalyviai pateikė (šių galimybių originali sistema neturėjo), patogiau

organizuota priimtų bylų saugojimo sistema.[5] Originali sistema leido pateikti ir vertinti trijų tipų uždavinius: [13]

- kai pradinių/galutinių duomenų srautas nukreipiamas iš/į ekrano;
- kai programa naudoja užduoties autorių pateiktą biblioteką, ir programos rezultatai pateikiami tik per biblioteką;
- kai vertinimui pateikiamos rezultatų bylos, o ne pati programa.

Lietuviška sistemos versija buvo papildyta tokiais uždavinių tipais:

- kai pradiniai/galutiniai duomenys yra byloje;
- kai pradiniai duomenys skaitomi iš bylos, o galutiniai rašomi ekrana;
- kai nėra pradinių duomenų, o rezultatas rašomas į bylą;
- kai programa naudoja užduoties autorių pateiktą biblioteką, ir programos rezultatai rašomi į bylą;
- kai pateikiamas teorinio uždavinio sprendimas (arba tik idėjos aprašas).

Viena didesnių problemų naudojant PVAS Lietuvos informatikos olimpiadose – saugumas. Svarbu nesudaryti galimybės nesąžiningai pasinaudoti sistema (pvz. prisijungti prie sistemos susikeitus su draugu slaptažodžiais). Todėl rekomenduojama varžybų metu prie sistemos prijungti tik vieną (ar kelis) kompiuterius ir stebėti kaip dalyviai naudojami sistema.

### **3.2.3. Virtualios mokymosi aplinkos**

Siekiant užtikrinti funkcionalumą, nagrinėjamos panašios elektroninio mokymo sistemos, turinčios testavimo posistemės:

#### **WebCT**

WebCT - tai virtuali mokymosi terpė, skirta teikti mokymo ir mokymosi paslaugas internete.[1] Ji sukurta British Columbia Universitete (Kanada) ir plačiai naudojama pasaulio universitetuose ir kolegijose, papildomą informaciją galima rasti gamintojo svetainėje adresu <http://www.webct.com>.

WebCT įrankiai leidžia teikti interneto aplinkoje kursus pagal įvairius scenarijus, organizuoti dinamišką, interaktyvų mokymą ir mokymąsi. Kaip pateikti kursą internete, priklauso nuo kurso turinio, mokymosi medžiagos sudėtingumo ir dėstytojo kuratoriaus pasirinktos teikimo metodologijos ir studentų paramos sistemos.

Į kursą gali būti įterptos šios priemonės: kurso turinio, ryšio, vertinimo bei studijų priemonės, Iš turinio priemonių galima paminėti: SD-ROM suteikiama galimybė prieiti prie informacijos kompaktiniame diske tiesiog iš savo kompiuterio, kurso užrašai, kalendorius; ryšio – pokalbiai, diskusijos; vertinimo priemonės – pavidimai (Assignments) leidžia

peržiūrėti atsiskaitymui skirtus dokumentus; pažymiai (My Grades) čia pateikiami testų vertinimai ir galutinis įvertinimas

### **eTest**

Ši programa yra *FirstClass* [2] sistemos papildomas įrankis, kurį galima įdiegti iš internetinio puslapio <http://www.jellybeansoftware.com>.

Ji skirta testų kūrimui naudojant specialiai tam reikalui sukurtas formas. Taip pat numatytas sukurtų testų laikymas. Išnagrinėjus šią sistemą buvo pastebėti keli trūkumai:

- Rezultatai pateikiami iš karto, tik pabaigus testo laikymą. Tai nėra priimtina, kadangi tokiu atveju studentai galėtų pasidalinti teisingais atsakymais su savo bendramoksliais;
- Šioje sistemoje žmonių, galinčių laikyti testą sąrašas yra sunkiai sudaromas. Jame galima įtraukti studentus pagal vartotojo vardą, bet tai nėra patogu, kadangi, esant dideliame studentų skaičiui, toks studentų suvedimas užimtų labai daug laiko. Antras sąrašų sudarymo būdas – įdėti programą į tam skirtą konferenciją. Tačiau šis būdas nepriimtinas tuo, jog tokį testą galėtų laikyti visi studentai, matantys tą konferenciją. Yra galimybė riboti studentus, galinčius matyti konferencijas, tačiau šį darbą gali atlikti tik sistemos administratorius;
- Sistema nepritaikyta lietuviškiems rašmenims.

### **Livelihood for Learning Management**

Šis „OpenText corporation“ produktas [3] yra pranašesnis už anksčiau paminėtus WebCT ir eTest, kadangi tai yra pilnai paruošta programa ne tik studentų žinių patikrinimui, bet ir mokomosios medžiagos platinimui. Ši sistema suteikia savo vartotojams šias galimybes:

- Kurti ir redaguoti mokymo kursus ir praktinius užsiėmimus
- Peržiūrėti pasirinktus kursus, įkelti bei išsaugoti įvairią medžiagą, susijusią su pasirinktu kursu.
- Kurti ir naudoti įvairius žinių patikrinimo testus ir apklausas.
- Kurti bendrus projektus.
- Bendrauti tarpusavyje.
- Pateikti automatiškai paskaičiuotus testo rezultatus
- Pateikti viso kurso kalendorių

Šios sistemos testavimo posistemę sudaro tik vieno tipo klausimai su keliais atsakymo variantais.



## **Colleen**

Ši nuotolinio mokymo programa padaryta kaip internetinė svetainė, visi veiksmai vykdomi kaip paprasčiausioje svetainėje. Nors pirmajame sistemos puslapyje susidarė įspūdis, jog ji yra integruota į FirstClass, tačiau FirstClass sistemoje yra saugojama tik nuoroda į nuotolinio mokymo sistemos svetainę. Dėl informacijos išdėstymo ir jos prieinamumo ši sistema yra tikrai patraukli, bet kaip ir Livelink for Learning Management testavimo sistema yra gana skurdi [4]. Testą sudarantys klausimai gali būti tik vieno tipo. Šis faktorius programos testavimo sistemą daro nepatrauklia. Be to sistema yra orientuota į skandinavų šalis, todėl angliška puslapio versija turi daug mažiau funkcijų ir galimybių nei švediškas variantas.

## **TestTool 4**

**TestTool4** - nuotolinio testavimo sistema, leidžianti kurti ir per internetą pateikti tekstinius ir grafinius testus. Testą sudarantys klausimai gali būti 4 tipų: vieno ir kelių atsakymų išrinkimo; grafinis klausimas; grafinių komponentų manipuliacija (algoritmų modeliavimas); programos teksto analizė. TestTool aplinkoje klausimas pateikiamas ne tik tekstine forma: yra galimybė manipuliuoti grafiniais bei daugialypės terpės (multimedijos) objektais, tai reiškia, kad klausimą gali sudaryti tokie elementai kaip garsų rinkmenos, valdantieji elementai (pvz.: teksto įvedimo ir pasirinkimo laukai), o tai sistemą daro gana patrauklią. Sistema sukurta panaudojant šiuolaikines programavimo technologijas, mokymo objektai saugomi XML failuose, o tai reiškia kad ateityje tokią sistemą bus galima suderinti su EMS standartais, palaikančiais XML technologiją.

### **3.2.4. EMS apžvalgos rezultatai**

Programavimo įgūdžiams testuoti yra svarbūs tiek teorinių žinių tiek jų praktinio pritaikymo galimybės. Programavimo varžybų testavimo sistemos ir PVAS skirtos tik praktinei realizacijai. Pirmojoje sistemoje sprendimai buvo įrašomi į diskelius arba paliekami darbiniam kataloge ir pasiekiami per tinklą. T. y. visi sprendimai (net ir nieko neskaičiuojančios programos) būdavo priimami. Tačiau naudojant PVAS atsirado galimybė dalyviui surasti jo paties sprendime kai kurias klaidas ir varžybų metu jas ištaisyti. PVAS yra tinklinė, dalyviai ją pasiekia naudodamiesi vietiniu tinklu ar internetu. Testuojama vertinimo serveryje ir tik priimtus darbus be to vertinimui pateikiama rezultatų bylos, o ne pati programa.

Universalios testavimo sistemos yra įkomponuotos daugumoje nuotoliniam mokymui skirtų programinės įrangos komplektų, laisvai platinamos Internetu, tačiau jos gerai tinka tiktai faktinių žinių kontrolei ir stokoja loginio mąstymo ir gebėjimo spręsti įvairius uždavinius kontrolės priemonių. Kaip buvo pastebėta, dauguma šių testavimo sistemų dažniausiai apsiriboja apklausos testais su pateikiamais galimų atsakymų rinkiniais, kurių programinis

realizavimas nesudaro problemų. Šis faktorius testavimo sistemą daro nepatrauklia. Todėl programavimo įgūdžių kontrolei šių testavimo sistemų panaudojimo galimybės taip pat labai ribotos.

Ir dar labai svarbi sistemų savybė - lankstumas arba integracijos galimybė. Pasirinkus nelanksčią sistemą, pastaroji gali tapti naujų galimybių panaudojimo stabdžiu. Apžvelgus elektroninio mokymo sistemas, turinčias testavimo posistemas, galima teigti, kad dauguma jų buvo kuriamos atskirais moduliai, vėliau juos apjungiant į bendrą vieningą visumą. Toks projektavimo būdas leidžia nesunkiai integruoti papildomus modulius ir funkcijas į funkcionuojančią sistemą. Be to kursams (testams) sudaryti dažniausia naudojama metakalba, sukurta XML technologijos, skirtos medžiagai aprašyti, pagrindu. Tai reiškia, kad šių kursų (testų klausimų) turinys yra aprašomas struktūriškai ir ateityje bus galima jį suderinti su kuriamais EMS kūrimo standartais (IMS, LOM, SCORM);

Pasaulyje yra sukurta daug mokymo sistemų, kurios nėra standartinės (mano nagrinėtos aplinkos) ir tam, kad jos leistų naudotis šiuolaikinio elektroninio mokymo privalumais, jas reikia standartizuoti. Todėl ir programavimo įgūdžių testavimo sistemoje pravartu spręsti standartizavimo klausimus.

3.1 lentelė

## Testavimo sistemų palyginimas

Pavadinimas		Vertinimo kriterijai					Atitikimas EM standartus
		Testo klausimų tipų realizavimas	Lankstumas arba integracijos galimybė	Saugumas			
				Duomenų saugumas	Tinklo saugumas	Išsilaužimo saugumas	
Testavimo sistemos	Testavimo sistemos, skirtos olimpiadoms	Tik praktinės užduotys	-	+/-	+/-	+/-	-
	Programavimo varžybų aptarnavimo sistemos (PVAS)	Tik praktinės užduotys	+/-	+	+	Sistema paleidžiama <u>tik nustatytu laiku</u> +	-
EMS, turinčios testavimo posistemas	eTest	1 tipo klausimai	+	+	+	+	-
	WebCT	1 tipo klausimai	+	+	+	+	-
	Colleen	1 tipo klausimai	+	+	+	+	-
	TestTool4	4 tipų klausimai	+	+	+	+	yra paruoštas standartizavimo modelis

1 tipo klausimai - tai apklausos testais su pateikiamais galimų atsakymų rinkiniais.

### 3.3. Elektroninio mokymo standartai

Taigi pasaulyje egzistuoja daugybė programinės įrangos priemonių, leidžiančių kurti mokymo medžiagą, testuoti žinias, tačiau dauguma jų neužtikrina bendro mechanizmo, kuris teiktų efektyvią mokymo išteklių paiešką ir prieinamumą, daugkartinį turinio ir metodų panaudojimą, užtikrintų suderinamumą. Ateityje perspektyvą turės tik tos EMS, kurias bus galima standartizuoti, o naujos kuriamos iš karto atsižvelgiant į organizacijų, kuriančių elektroninio mokymo standartus nurodymus. Plačiausiai naudojami IMS, IEEE LOM, Dublin Core, ADL SCORM.

**IMS.** *IMS Global Learning Consortium, Inc.* kuria atviras specifikacijas tam, kad palengvintų tinkle esančius mokymosi darbus, tokius kaip mokymo medžiagos suradimas ir naudojimas, besimokančiojo progreso sekimas, mokinio atliktų užduočių įvertinimas, apsikeitimas studijuojančiųjų duomenimis tarp administracinių sistemų.

IMS teikia pirmenybę apibrėžimui ir palaikymui techninių specifikacijų, kurios skirtos patenkinti nuotolinio mokymo platintojų poreikiams. Šios specifikacijos suteikia galimybę kurti mokymo produktus ir paslaugas, pagrįstus bendru suderinamumo (visur veikimo) principu.

Nuotoliniame mokyme yra daugybė sričių, kurios reikalauja visur veikimo. IMS atsižvelgė į daugumą jų, o likusias specifikuos ateityje. IMS apima šias sritis[12]:

- IMS mokymo resursų metaduomenų specifikacija. Skirta mokymo resursams aprašyti, kad palengvintų paiešką ir suradimą;
- IMS verslo specifikacija. Skirta dalintis duomenimis apie besimokančiuosius, kursus, atlikimą ir kt. administracinių programų pagalba;
- IMS mokomosios medžiagos paketų specifikacija. Skirta pakartotinio panaudojimo objektams kurti ir dalijimuisi jais tarp vartotojų;
- IMS klausimų ir testų specifikacija. Skirta testavimo ir kitokių vertinimo įrankių dalijimuisi tarp vartotojų;
- IMS besimokančiojo informacijos paketo specifikacija. Skirta besimokančiojo informacijos tvarkymui taip, jog mokomoji sistema kuo labiau tenkintų specifinius vartotojo poreikius;
- IMS pakartotinio panaudojimo kompetencijos apibrėžimo specifikacija. Skirta apibūdinti, charakterizuoti ir keisti esmines kompetencijos charakteristikas;
- IMS paprasto eilės nustatymo specifikacija. Skirta nustatyti, kaip mokymo objektai surūšiuojami ir koku būdu pateikiami besimokančiajam;

- IMS prieinamumas (ši sritis gali būti niekad nespecifikuota). Skirta suteikti gaires kitoms su IMS dirbančioms grupėms, tam kad būtų užtikrinta, jog kiekviena specifikacija garantuoja maksimalų prieinamumą bet kokių gebėjimų besimokančiajam;
- IMS mokymo plano specifikacija. Skirta apibrėžti mokymo scenarijus ir sąveikas mokomosios medžiagos arba kursų kūrėjams;
- IMS skaitmeninių saugyklų specifikacija. Skirta integruoti nuotolinį mokymą su svarbiais informacijos šaltiniais.

**ADL SCORM** (*Sharable Content Object Reference Model*). SCORM specifikacijos buvo sukurtos *Advanced Distributed Learning (ADL)* kompanijos iniciatyva. Prieš SCORM sukūrimą elektroninio mokymo medžiagos nebuvo galima pernešti viso kurso arba atskirų jo dalių iš vienos elektroninio mokymo sistemos į kitą. Taigi galima sakyti, jog nebuvo galimas ir apsikeitimas mokymosi medžiaga.

Tam, kad suprasti, kas tai yra SCORM, reikėtų pažvelgti į SCORM kursą arba jo dalį. Jie yra pateikiami kaip paketai specialiu formatu PIF (*Packet Interchange Format*). Tokio paketo pavyzdys – zip formato failas, kuriame yra failai su informacija apie turinį, turinio metaduomenis, kurso metaduomenis ir kurso struktūra (turinys su navigacijos ir sekų informacija). Kad šis paketas būtų visur veikiantis, reikia jog bent viename iš turinio failų būtų metodai, kurie suteiktų sąsają su bet kuria elektroninio mokymo sistema.

SCORM sudaro dvi specifikacijos: turinio sukaupimo modelis ir veikimo aplinka. Turinio sukaupimo modelis aprašo, kaip apibūdinama individuali mokomoji medžiaga ir kaip iš mokomosios medžiagos sukurti visiems prieinamą ir visur veikiantį kursą. Veikimo aplinka aprašo taikomųjų programų programavimo sąsają (API) ir duomenų modelį kurso paketų sąveikai su elektroninio mokymo sistema.

SCORM yra kuriamas kaip atskirų specifikacijų apjungta biblioteka. Taip stengiamasi apjungti pasaulyje priimtus standartus į visumą, kad programinės įrangos kūrėjams būtų paprasčiau kurti el. mokymo sistemas, kurios atitinka visus svarbiausius standartus. Savaimė suprantama, jog tokie SCORM kūrėjų užmojai susilaukia ir kritikos, nes šis standartas yra pastoviai tobulinamas ir plečiamas. Štai viename straipsnyje „*Is SCORM coming out ahead?*“ („Ar SCORM yra žingsnis į priekį?“) [7], autorius pateikia ne tik gerąsias, bet ir blogąsias SCORM puses:

- *Gerosios*:
  - Pati geriausia savybė yra ta, jog mokymo medžiaga sukurta su SCORM palaikančia sistema, bus panaudojama ir kitoje EMS palaikančioje SCORM standartą;

- Yra daug atskirų standartų, o SCORM apima visa tai, kas yra būtina;
- SCORM suteikia efektyvias galimybes kurti mokymo kursus ir jų pagalba vertinti besimokančiuosius. Taigi net jeigu SCORM nepavyktų pasiekti savų tikslų, jis bent šiuo metu mokomosios medžiagos kūrėjams suteikia atspirties tašką.
- *Blogosios:*
  - Pagrindinė bloga savybė yra ta, jog SCORM įdiegimas yra komplikuoatas ir savo ruožtu daugiau apima labai paprastas architektūras;
  - SCORM sudėtingumas atsirado dėl to, jog jis nukrypo nuo savo pirminio tikslo. Pradžioje SCORM buvo sumanytas apibrėžti mokymo medžiagą, kad ją būtų galima pernešti iš vienos sistemos į kitą. Tačiau dabar SCORM apima ir pačios el. mokymo sistemos standartizaciją.
  - Autorius teigia, jog SCORM nelabai gali būti laikomas standartu, nes dar vis nėra pasiektas konsensusas dėl jo. Naujose SCORM versijose pridedamos naujos specifikacijos, jos gali destabilizuoti standartą

**Dublin Core** standartas apima 15 elementų (pavadinimas, autoriai, raktiniai žodžiai, leidėjas, data, tipas, formatas, identifikatorius, šaltinis, kalba, ryšiai, apimtis, teisės), kurie naudojami įvairiausiems ištekliams tinkle aprašyti. Kiekvienas elementas yra nebūtinai ir gali būti kartojamas, jų eiliškumas neturi reikšmės. Kai kurių elementų turinio duomenys gali būti parenkami iš žodynų, turinčių tam tikrą dažnai naudojamų terminų bei sinonimų aibę.

Dublin Core pagrindu buvo sukurtas IEEE LOM – mokymo objektų meta modelis

**IEEE LOM** (*IEEE Learning Object Metadata-Data Model*). 2002 metų birželio 12 d. šis standartas oficialiai buvo patvirtintas IEEE standartų asociacijos. IEEE mokymo technologijų standartų komitetas mokymo objekto metaduomenų (LOM) standartą pradėjo kurti nuo 1997 metų. Šiame procese dalyvavo ir tebedalyvauja asmenys, atstovaujantys 15 šalių. Tai yra pirmasis įvairiapusiškas LOM standartas, dėl kurio kūrimo, palaikymo ir tolimesnio vystymo yra atsakinga LTSC (*Learning Technology Standards Committee*).

Šis standartas yra skirtas mokymo objektų paieškai, įvertinimui, išmokimui ir naudojimui palengvinti. Taip pat vienas iš tikslų yra palengvinti prieinamumą prie mokymo objektų ir paprastesnį apsikeitimą jais atsižvelgiant į kultūrinius ir kalbinius skirtumus tarp įstaigų, kuriose mokymo objektai bus naudojami. Standarto tikslas yra aprašyti bendrą schemą, kuri galėtų būti naudojama praktikoje, pvz. palengvinti automatizuotą mokymo objektų tvarkymą, kurį atliktų programinės įrangos agentai.

Standartas neapibrėžia, kaip elektroninio mokymo sistema atvaizduos arba naudos metaduomenų reikalavimus mokymo objekte.

### 3.3.1. Metaduomenų standartai

Efektyviam mokymo išteklių organizavimui standartų organizacijos siūlo naudoti metaduomenis, kurie suprantami kaip informacijos išteklius aprašantys įrašai, naudojami įvairiems tikslams, pvz išteklių, kurie turi reikalingą informaciją, identifikavimas, jų tinkamumo naudoti įvertinimas ir panašiai. Metaduomenys gali būti saugomi tiek dokumente, kurį jie aprašo, tiek bet kokioje duomenų bazėje. Bendra metaduomenų struktūra:

- 1) *General* – bendra informacija aprašanti mokymo objektą kaip visumą;
- 2) *Lifecycle* – mokymo objekto istorija ir esama būsena bei informacija apie tuos, kurie įtakojo šio objekto evoliuciją;
- 3) *Meta-Metadata* – informacija apie pačius metaduomenis;
- 4) *Technical* – mokymo objekto techniniai reikalavimai ir charakteristikos;
- 5) *Educational* – mokymo objekto edukacinės ir pedagoginės charakteristikos;
- 6) *Rights* – mokymo objekto naudojimo sąlygos ir intelektualinės nuosavybės teisės;
- 7) *Relation* – aprašo sąryšį tarp šio ir kitų mokymo objektų;
- 8) *Annotation* – komentarai apie tai kaip šį objektą naudoti mokymesi ir kas parašė šiuos komentarus;
- 9) *Classification* – apibūdina mokymo objektą pagal tam tikrą klasifikavimo sistemą.

Metaduomenys yra aprašomi XML pagalba. Siekiant palengvinti resursų kūrėjo darbą, aukštos kokybės sistemose yra sukurti specialūs įrankiai, padedantys aprašyti metaduomenis. Toks įrankis padeda nepasiklysti tarp daugybės metaduomenų, užtikrina jog visi būtini elementai būtų aprašyti, taip pat iš karto galėtų atlikti validaciją ar teisingai suformuoti metaduomenys. Automatizuotas įrankis tai gali atlikti pats, patikrindamas XML failą pagal standartų kūrėjų paruoštus .DTD arba .XSD failus. Šiuose failuose aprašytų schemų dėka yra patikrinama ar yra aprašyti visi būtini elementai, ar nėra neleistinių XML žymių ir ar teisingai suformuotas pats XML failas. Tai galima padaryti XML redaktorių pagalba (pvz. *XML Spy*) arba ADL organizacijos sukurtu specialiu el. mokymo medžiagos testavimo įrankiu „*SCORM Conformance Test Suite VI.2.3*“. Tai nemokamas įrankis. Jį parsisiųsti galima iš ADL svetainės, adresu

<http://www.adlnet.org/index.cfm?fuseaction=scormdown&cfid=547391&cftoken=18875176>. Metaduomenų standartai dažniausia taikomi tiems informacijos ištekliams, kurie bus prienami internete, siekiant palengvinti paiešką, išvaduoti vartotoją nuo perteklinės informacijos.

### 3.3.2. Mokymo objektų standartai

Elektroninio mokymo sistemoje smulkiausia mokymo medžiagos dalis yra kažkokie tai resursai. Tai paprasčiausi .doc, .xml, .wav, .jpeg, .mov, .gif, .html, .avi ir pan. failai. Tokių resursų rinkinys susijęs su el. mokymo sistema sudaro mokymo objektą. Mokymo objektų pavyzdžiais gali būti daugiaterpės aplinkos turinys, mokymo turinys, mokymo taikomoji programinė įranga. Plačiaja prasme mokymo objektai gali apimti mokymo tikslus, asmenis, organizacijas, įvykius.

Šie objektai privalo turėti bent du kreipinius į taikomųjų programų programavimo sąsają (API): `LMSInitialize(“”)` ir `LMSFinish(“”)`. Mokymo medžiagos objektas gali būti aprašytas naudojant paprastą teksto ar HTML redaktorių.

Mokymo objektą galima kurti kaip paprasčiausią HTML failą. Šiuo atveju tarp `<HEAD>` žymių aprašomos funkcijos, kurios vykdo kreipimąsi į el. mokymo sistemos API sąsają, bei visos kitos papildomos funkcijos, kurios reikalingos kuriamo objekto valdymui. Funkcijos dažniausia rašomos *java script* kalba. Pagrindiniai 2 kreipiniai į funkcijas gali būti realizuotos tokiomis SCORM (standartas)siūlomomis funkcijomis:

#### 3.1 lentelė

#### Mokymo objektų funkcijos realizuojančios būtinus kreipinius į API sąsają

<i>Kreipinys į LMSInitialize(“”)</i>	<i>Kreipinys į LMSFinish(“”)</i>
<p>Funkcija <i>doLMSInitialize</i>. Įėjimo duomenų nėra. Gražinama reikšmė <code>CMIBoolean = TRUE</code>, jei kreipimasis į mokymo sistemos API sąsają buvo sėkmingas, priešingu atveju - <code>CMIBoolean = FALSE</code>. Ši funkcija pradeda darbą su el. mokymo sistema, kreipdamasi į <code>LMSInitialize(“”)</code> funkciją, kuri turi būti įdiegta sistemos pusėje.</p>	<p>Funkcija <i>doLMSFinish</i>. Įėjimo duomenų nėra. Gražinama reikšmė <code>CMIBoolean = TRUE</code>, jei funkcija įvykdyta sėkmingai, nesėkmės atveju <code>CMIBoolean = FALSE</code>. Ši funkcija uždaro sąsają su el. mokymo sistema iškviečiant <code>LMSFinish(“”)</code> funkciją, idiegtą sistemos pusėje.</p>
<pre>function doLMSInitialize() { var api = getAPIHandle(); if (api == null) { alert( "Unable to locate the LMS's API Implementation.\n" + "LMSInitialize was not</pre>	<pre>function doLMSFinish() { var api = getAPIHandle(); if (api == null) { alert( "Unable to locate the LMS's API Implementation.\n" + "LMSFinish was not successful.");</pre>

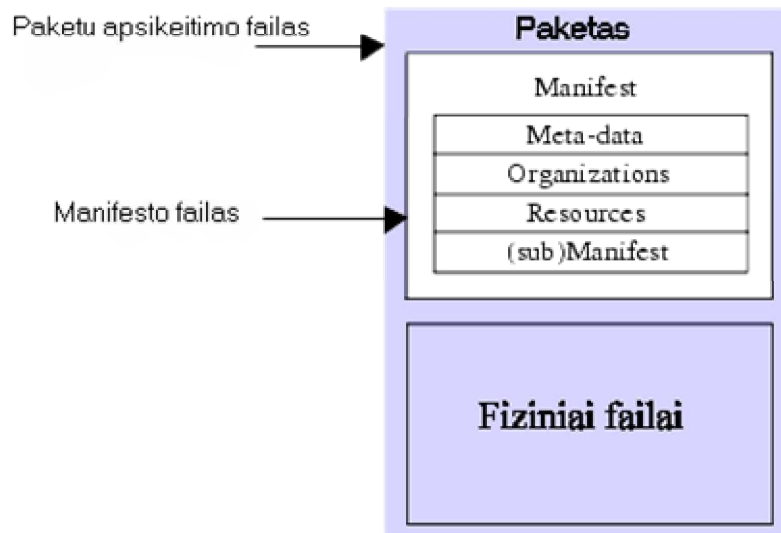
<pre> successful.");         return "false";     }     var result = api.LMSInitialize("");     if (result.toString() != "true")     {         // čia galima pridėti klaidų         valdymą     }     return result.toString(); } </pre>	<pre>         return "false";     }     else     {         // Kviečiama LMSFinish funkcija, tik         jei ji nebuvo kviesta anksčiau         if ( ! finishCalled )         {             finishCalled = 1;             // Kviečiama LMSFinish funkcija             var result = api.LMSFinish("");             if (result.toString() != "true")             {                 // čia galima pridėti klaidų                 valdymą             }         }     }     return result.toString(); } </pre>
---	---

Lentelėje aprašytos funkcijos naudoja getAPIHandle() funkciją. Mokomosios medžiagos objekto kūrėjas prideda mokymo medžiagą (testavimo klausimus) įprastu būdu, tiesiog HTML dokumente aprašydamas ją. Galima mokomosios medžiagos pabaigoje panaudoti koks nors mechanizmą (pavyzdžiui mygtuką), kurio pagalba vartotojas nurodytų jog baigė šią medžiagą. Šis mechanizmas leistų žinoti kada vykdyti kreipinį į LMSFinish funkciją.

Formuoti mokymo objektų metaduomenis XML galima keliais būdais. Vienas iš galimų būdų – transformuoti egzistuojančios reliacinės duomenų bazės (RDB) turinį į XML ir atlikti paiešką bei standartinių dokumentų formavimą XML apdorojimo priemonėmis. RDB duomenų transformavimo į XML priemonės egzistuoja daugelyje naujų DBVS versijų, pavyzdžiui, MS SQL Server 2000, Oracle 8i. Kitas būdas – išgauti reikiamus duomenis RDBVS priemonėmis ir iš jų formuoti standartinius dokumentus.

Tam, kad visi aprašyti resursai ir mokymo objektai chaotiškai neišsimėtytų ir būtų aišku ką ir kada naudoti, aprašoma mokymo medžiagos paketo struktūra. Mokomosios medžiagos talpinimas į paketus yra susijęs su nuotolinės mokomosios medžiagos aprašymu, struktūra ir buvimo vieta. Mokymo medžiagos pakete privalo būti manifestas, kuriame yra aprašomi resursai ir jų struktūra..





**3.2 pav. IMS mokomosios medžiagos talpinimo į paketą modelis**

Paveiksle pavaizduotas IMS paketas susideda iš dviejų pagrindinių elementų: specialaus XML failo, kuris aprašo mokomosios medžiagos organizacinę schemą bei resursus pakete ir fizinių failų, apibūdintų XML. Specialusis XML failas vadinamas *IMS Manifest* failu, kadangi kurso turinys bei organizacinė schema yra apibūdinami manifestais. Kada paketas yra įdėtas į vieną failą pernešimui, jis yra vadinamas paketų apsikeitimo failu.

Paketų apsikeitimo failas – tai vienas failas (pvz. \*.zip, \*.jar, \*.cab formato), kuriame yra pagrindinis failas ir visi kiti fiziniai failai (manifestuose aprašyti laikmenų elementai, tekstiniai failai, piešiniai ir kitokie resursai išdėlioti tam tikruose pakatalogiuose). Paketų apsikeitimo failas yra glaustas perdavimo formatas pasauliniame tinkle. PKZip v2.04g (.zip) formatas yra rekomenduojamas kaip pagrindinis paketų apsikeitimo failo formatas.

Paketas – tai loginis katalogas, kuriame yra specialiai pavadintas XML failas bei jo valdomi bet kokie XML valdymo dokumentai (tokie kaip DTD ar XSD failai) ir pakatalogiai, kuriuose yra šaltiniai (mokymo medžiaga)

Pagrindinis manifestas – būtinas XML elementas, kuris apibūdina paketą. Jame taip pat gali būti ir papildomi manifestai. Kiekvieną manifestą sudaro tokios dalys:

- Metaduomenų dalis – XML elementas apibūdinantis manifestą;
- Struktūrinė dalis – XML elementas apibūdinantis vieną arba keletą manifesto mokomosios medžiagos struktūrų;
- Resursų dalis – XML elementas turintis nuorodas į resursus ir laikmenų elementus reikalingus manifestui. Šioje dalyje taip pat aprašomi resursų metaduomenys ir nuorodos į kitus išorinius failus;

- Papildomi manifestai – vienas arba keli neprivalomi, logiškai išdėstyti manifestai.

Paketas atitinka dalį panaudojamos mokymo medžiagos. Tai gali būti dalis kurso, naudojama nepriklausomai, visas kursas arba net keletas kursų. Kuomet paketas yra naudojamas kažkokioje elektroninio mokymo sistemoje, turi būti galima jį sujungti arba išskaidyti į kitus paketus. Paketas privalo būti savarankiškas, t. y. jame turi būti visa reikiama informacija jo panaudojimui po išpakavimo.

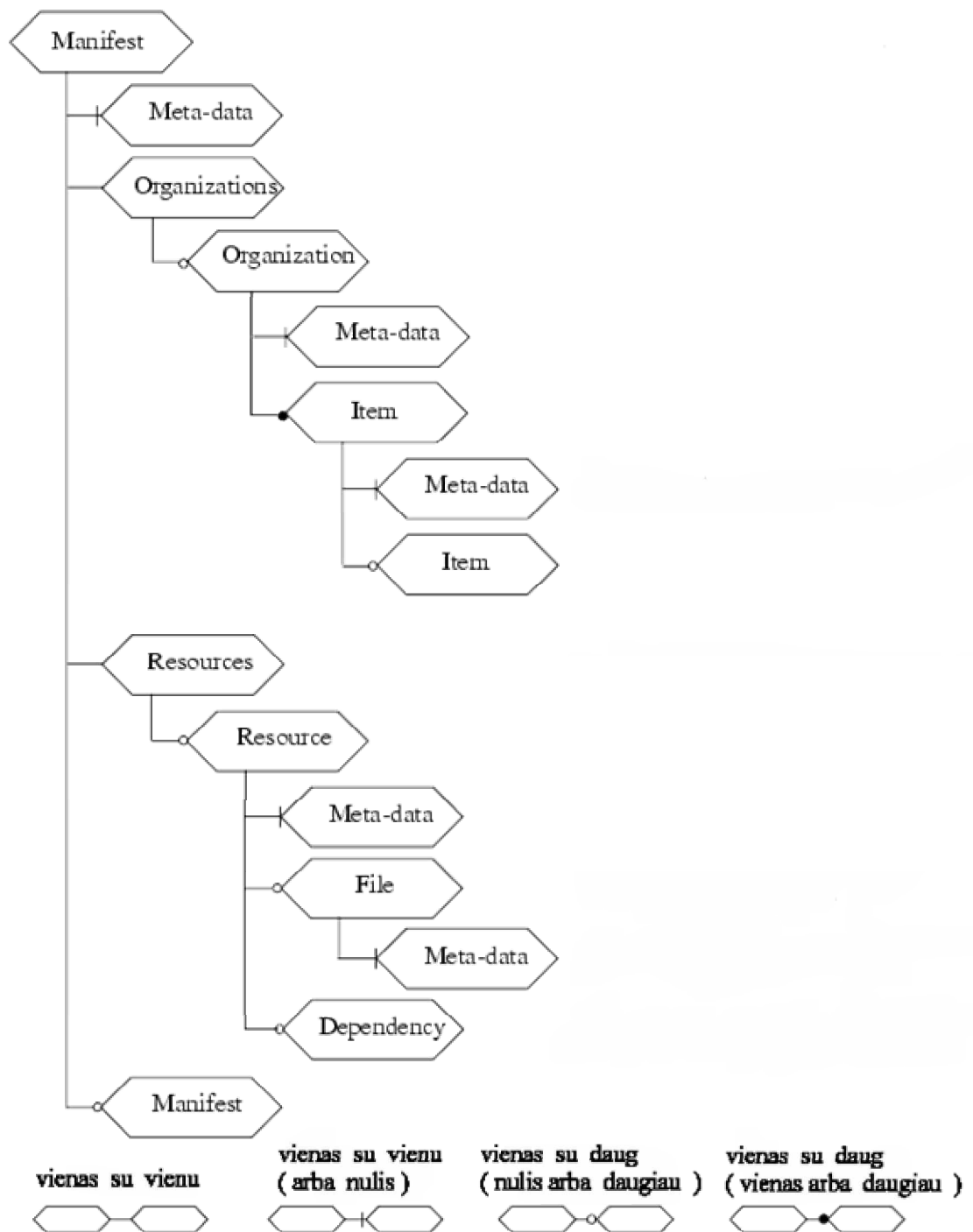
Nėra reikalaujama, jog paketas būtų sudėtas į paketų apsikaitimo failą. Paketas gali būti pernešamas nebūtinai suspaustas į vieną failą, o bet kokioje mobiliuoje terpėje (CD-ROM ir panašiai). Svarbiausia, jog pagrindinis IMS manifesto failas ir pagalbinių XML failai (DTD, XSD) būtų šakniniame kataloge.

Manifeste XML kalboje aprašomi resursai, sudarant reikšmines instrukcijas. Taip pat jame gali būti aprašyta nulis arba daugiau struktūrizavimo būdų pagal kuriuos atvaizduojami resursai. Pagrindinė taisyklė yra tokia, kad pakete turi būti vienas pagrindinis manifestas, kuriame gali būti vienas arba daugiau pagalbinių manifestų. Pagrindinis manifestas visuomet apibūdina paketą. Pagalbiniai manifestai yra skirti apibūdinti tam tikro hierarchinio lygio mokomąją medžiagą: kursą, instrukcijų objektą ar kita.

Pavyzdžiui, jei mokomoji medžiaga kurse yra taip glaudžiai susijus tarpusavyje, kad neįmanoma išskirti kažkokių atskirų dalių, šios medžiagos rengėjas naudos vieną manifestą kurso ir jo struktūros aprašymui. Tačiau mokomosios medžiagos kūrėjai, kurie ruošia mokomuosius objektus, kurie gali būti apjungti su kitais tokiais objektais, naudoja atskirą manifesto failą kiekvienam objektui. Po to šie manifestai apjungiami į vieną aukštesnio lygio manifestą, kuriame aprašoma kurso struktūra. Galiausiai, jei norima keletą kursų sujungiant į vieną paketą sudaryti mokymo programą, pagrindiniame manifeste bus po papildomą manifestą aprašyti kiekvienam kursui.

Manifeste aprašyti resursai gali būti: tinklalapiai, laikmenų failai, tekstiniai failai, vertinimo objektai ir kitokie duomenys failuose. Resursais taip pat gali būti duomenys, kurie nėra pakete, bet prieinami interneto pagalba arba resursų rinkiniai aprašyti pagalbiniuose manifestuose. Kiekvienas resursas manifesto faile aprašomas <resource> žyme. Šios žymės viduje aprašomi pakete esantys failai. Tam naudojama <file> elementas.

Sekančiame paveiksle pateikiami pagrindiniai Manifesto elementai.[11]



3.3 pav. Manifesto elementai [11]

Detalizuosiu kiekvieną šios schemos elementą, pateikiant ir papildomus smulkesnius elementus:

*Manifest* – pakartotinai panaudojama mokymo dalis. Apima *Meta-data*, *Organizations*, *Resources* elementus.

*Identifier* – identifikatorius, kuris yra unikalus konkrečiame manifeste.

*Version* – nurodo manifesto versiją.

*Meta-data* – metaduomenys apibūdinantys manifestą.

*Schema* – aprašoma schema, kuri apibrėžia ir valdo manifestą.

*SchemaVersion* – nurodo schemos versiją.

{IMS metaduomenys} – šioje vietoje yra įterpiami IMS metaduomenys.

*Organizations* – aprašo vieną arba kelias paketo struktūras.

*Default* – nurodoma kokia struktūros schema yra naudojama pagal nutylėjimą.

*Organization* – apibūdina tam tikrą hierarchinę struktūrą.

*Identifier* - identifikatorius, kuris yra unikalus konkrečiame manifeste.

*Structure* – pagal nutylėjimą yra reikšmė „*hierarchical*“. Skirtas nusakyti struktūros formą.

*Title* – aprašo struktūros pavadinimą.

*Item* – mazgas, kuris aprašo struktūros formą.

*Identifier* - identifikatorius, kuris yra unikalus konkrečiame manifeste.

*IdentifierRef* – nuoroda į identifikatorių resursų dalyje.

*Title* – *Item* pavadinimas.

*IsVisible* – nurodo ar rodomas šis elementas, kada atvaizduojamas paketas.

*Parameters* – statiniai parametrai, kurie paduodami resursams paleidimo metu.

*Item* – smulkesnis struktūros mazgas.

*Meta-data* – šį elementą (*Item*) apibūdinantys metaduomenys.

{IMS metaduomenys} - šioje vietoje yra įterpiami IMS metaduomenys.

*Meta-data* –struktūrą apibūdinantys metaduomenys.

{IMS metaduomenys} - šioje vietoje yra įterpiami IMS metaduomenys.

*Resources* – tai nuorodų į resursus rinkinys. Čia nekreipiamas dėmesys į eilės tvarką ar hierarchiją.

*Resource* – nuoroda į resursą.

*Identifier* - identifikatorius, kuris yra unikalus konkrečiame manifeste.

*Type* – resurso tipas.

*HRef* – nuoroda į internetinį adresą (URL).

*Meta-data* – resursą apibūdinantys metaduomenys.

{IMS metaduomenys} - šioje vietoje yra įterpiami IMS metaduomenys.

*File* – failų sąrašas, su kuriais susijęs šis resursas.

*HRef* – nurodo failo vietą.

*Meta-data* – šį failą apibūdinantys metaduomenys.

{IMS metaduomenys} - šioje vietoje yra įterpiami IMS metaduomenys.

*Dependency* – nurodo resursą, su kurio failais susijęs šis resursas.

*IdentifierRef* – nuoroda į identifikatorių resursų dalyje.

Aprašius struktūros dalį į manifestą gali būti įdėti mokomosios medžiagos paketo metaduomenys - aprašoma schema ir jos versija, bei pridedami metaduomenys.

Mokomosios medžiagos paketas išreiškia duomenų struktūrą, kuri leidžia užtikrinti pakartotinį panaudojamumą ir visur veikimą tarp nuotolinio mokymo medžiagos, mokymo medžiagos kūrimo įrankių, skaitmeninių saugyklų, elektroninio mokymo sistemų, realaus laiko sistemų ir pan.

Sukurti paketą nesunkiai galima rankiniu būdu. Tačiau būtų naudinga jei el. mokymo sistemoje būtų įrankis, skirtas failų sistemos valdymui. Toks įrankis padėtų užtikrinti, kad visi manifeste aprašyti failai sudarytų teisingą struktūrą.

### 3.3.3. Standartai mokymo medžiagos tikrinimui

Standartų kūrėjai pateikia .DTD ir .XSD formatų failus, tam kad patikrinti ar kuriama mokymo medžiaga atitinka visas specifikacijoje numatytas taisykles. XML redaktoriaus pagalba galima patikrinti ar sukurtas XML failas yra teisingai suformuotas ir atlikti jo validaciją pagal tam tikro standarto XML schemą. Tokia validacija turėtų būti įdiegta el. mokymo sistemoje ir vykdoma kiekvienoje mokomosios medžiagos sudarymo stadijoje.

Naudojant specialias programas galima patikrinti ar sukurta medžiaga atitinka DTD aprašymą. ADL organizacija yra sukūrusi specialų įrankį „*SCORM Conformance Test Suite VI.2.3*“, kuris atlieka keturis pagrindinius mokymo sistemos testus. Šį įrankį nemokamai galima parsisiųsti iš ADL interneto svetainės, adresu:

<http://www.adlnet.org/index.cfm?fuseaction=scormdown&cfid=547391&cftoken=18875176>.

„*SCORM Conformance Test Suite VI.2.3*“ atliekami testai:

- Elektroninio mokymo sistemos veikimo aplinkos testas (*Learning Management System (LMS) Run-Time Environment Conformance Test*). Jis skirtas el. mokymo sistemos kūrėjams, patikrinti ar kuriama sistema atitinka SCORM RTE standartą. Kitais žodžiais tariant, tikrinama ar el. mokymo sistemos API sąsaja yra teisinga.

- Mokymo objekto veikimo aplinkos testas (*Sharable Content Object (SCO) Run-Time Environment Conformance Test*). Šio testo pagalba tikrinama ar teisinga mokymo objekto API sąsaja. Natūralu, jog jei tiek el. mokymo sistema, tiek mokymo objektas turi teisingas API sąsajas, objektas bus panaudojamas mokymo sistemoje.
- Metaduomenų atitikimo testas (*Meta-data Conformance Test*). Atliekamas metaduomenų tikrinimas pagal SCORM turinio surinkimo modelį. Tikrinti galima resursų, mokymo objektų ir mokymo medžiagos paketo metaduomenis.
- Mokymo medžiagos paketo atitikimo testas (*Content Package Conformance Test*). Tikrinama ar suformuotas paketas atitinka mokymo medžiagos talpinimo į paketus standartą. Dar prieš šį testą turi būti sukurti visi mokomosios medžiagos paketo failai ir kiekvienas jų patikrintas atskirai. Mokymo medžiagos paketo atitikimo testas patikrina tiek fizinę paketo struktūrą, tiek ir mokymo objektus su metaduomenimis, kurie yra pakete.

Kiekvieno testo metu yra sugeneruojamas failas, kuriame surašomi atlikti veiksmai ir aptiktos klaidos. Atsižvelgiant į aptiktas klaidas reikia taisyti mokymo medžiagą ir jos kūrimo įrankius, kad tokios klaidos nepasikartotų kitą kartą.

### **3.4. Elektroninio mokymo perspektyva**

Naujausios elektroninio mokymo sistemos (EMS) yra kuriamos iš karto atsižvelgiant į pasaulinius standartus (IEEE LOM; IMS, ADL SCROM), kad vartotojai galėtų prisijungti prie elektroninių saugyklų, kuriose saugomi mokymo objektai bei mokomoji medžiaga ir iš jų pasiimti jiems reikiamus mokymo resursus. (Senos EMS išliks tik tos, kurias bus galima standartizuoti.) Turi būti galimybė paimti mokymo komponentus sukurtus su vienokiais įrankiais ar platforma ir naudoti tuos pačius komponentus kitoje vietoje su skirtingais įrankiais ar platformomis

Numatoma, jog bus sukurta mokymo žinių bibliotekos arba saugyklos, kuriose bus kaupiami mokymo objektai. Šie objektai turės būti bet kuriuo metu prieinami interneto pagalba. Pilnai įmanoma, jog tokių saugyklų sukūrimas bus pagrindas naujos rūšies ekonomikai, kai mokomosios medžiagos kūrėjai gaus pinigus už sukurtus aukštos kokybės mokymo objektus. Taip pat tai turėtų paskatinti naujų paslaugų ir produktų atsiradimui. Šie savo ruožtu suteiks lengvai prieinamus, pritaikomus ir tarpusavyje dalinamus mokymosi būdus. Kada yra prikurta prieinamų mokymo objektų, juos bet kada galima sujungti realiam laike ir pateikti besimokantiems, kai tik to reikia. Todėl ir šio projekto realizacijoje reikėtų spręsti standartizavimo klausimus.

## **4. PROGRAMAVIMO ĮGŪDŽIŲ TESTAVIMO SISTEMOS PROJEKTAVIMAS**

Pasirinkta objektinė projektavimo metodika, nes kaip parodė analizės rezultatai dauguma EMS kuriamos atskirais moduliais, vėliau juos apjungiant į bendrą vieningą visumą. Toks projektavimo būdas leidžia nesunkiai integruoti papildomus modulius ir funkcijas į funkcionuojančią sistemą. Tai gi objektinis projektavimas supaprastina sistemos programinę realizaciją, išplečia bendras sistemos funkcines galimybes.

Programinės įrangos projektavimo procese labai svarbus yra ir prototipų vaidmuo. Prototipai padeda išsiaiškinti papildomus sistemos funkcinius ir nefunkcinius reikalavimus, pateikia pradinį sistemos vaizdą. Todėl projektui realizuoti bus naudojama prototipo kūrimo metodika. Šis metodas pasirinktas, nes dar kūrimo procese vartotojas gali susipažinti su veikiančia sistema, pamatyti jos galimybes, aptikti klaidas ir praleistus reikalavimus.

### **4.1. Reikalavimai programavimo įgūdžių testavimo sistemai**

Norint sukurti lanksčią programavimo įgūdžių sistemą, pirmiausiai reikia gerai išsiaiškinti ko norima iš tokios sistemos, ką ji turi atlikti, kokiomis savybėmis pasižymėti.

Atlikus kompiuterinių testavimo sistemų analizę, galima daryti išvadą, kad kompiuterinė programavimo žinių testavimo sistema turi pateikti vartotojams tiek žinių kontrolės administravimo, tiek įvairių tipų testų parengimo bei jų rezultatų įvertinimo priemonės ir savaime suprantama turi būti visuotinai priimtų standartų palaikymas.

Atitinkami reikalavimai sistemai yra keliami testuotojo požiūriu:

- registracija sistemoje (identifikacijai naudojamas vartotojo vardas ir slaptažodis);
- testų sudarymui skirta aplinka;
- naujų vartotojų kūrimas bei pašalinimas;
- testų parametrų (užduočių skaičiaus, testo trukmės, tikrinimų skaičiaus) parinkimas;

Testuojamojo požiūriu sistemoje turi būti įgyvendintos sekančios funkcijos:

- registracija sistemoje;
- užduočių atlikimui skirta aplinka .

#### **4.1.1. Reikalavimai programavimo įgūdžių kontrolės užduočių tipams**

Automatinių testavimo sistemų efektyvumas ir praktinio taikymo galimybes priklauso nuo jose realizuojamų užduočių tipų įvairovės, todėl šias sistemas reikia parengti taip, kad jos galėtų aptarnauti

bent keletą konkrečiai taikomajai sričiai pritaikytu užduočių tipų. Aprašomoje programavimo įgūdžių kontrolės sistemoje pakaktų realizuoti trijų tipų testų:

- apklausos; (vieno ar kelių atsakymų pasirinkimas)
- programų tekstų analizės,
- programų taisymo ir rašymo.

Tradiciniai apklausos testai vartojami tikrinant, kaip įsisavintos pagrindines sąvokos ir žinios apie programavimo kalbos struktūras, o taip pat pradiniam studentų susipažinimui su sistemos darbu.

Sudėtingesniais analizės testais, kurie yra apklausos testų atmaina, jau galima patikrinti ne tik tai išmoktas sąvokos, bet ir sugebėjimą pritaikyti įgytas žinias įvairių situacijų analizei. Šiems testams būdinga tai, kad juose pateikiami ne tik klausimai ir galimu atsakymu rinkiniai, bet ir iššios programos arba jų fragmentai, kurie iliustruoja konkrečią programavimo situaciją.

Programų taisymo bei rašymo testai, kuriuose testuojamajam leidžiama keletą kartu pasinaudoti kompiliatoriaus paslaugomis ir patikrinti, ar teisingai atlikta testo užduotis.

Nemažai problemų kyla tikrinant programavimo uždavinių sprendimo rezultatus, nes tai yra ne tikslios reikšmės, o programų tekstai, kuriuos galima patikrinti tik specialioje konkrečiai programavimo kalbai pritaikytoje verifikavimo aplinkoje. Tokią aplinką galima sukurti papildant testavimo sistemą jos valdomu programavimo kalbos kompiliatoriumi. Taip sudarytoje testavimo sistemoje yra griežti reikalavimai užduočių struktūrai:

- rezultatas privalo būti iššisa vykdymui kompiuteryje parengta programa.

Tokių testų užduotys sudaromos iš keturių dalių:

- pačios užduoties aprašymo,
- komentarų bei papildomų nurodymų testuojamojo rašomam programos fragmentui,
- sudaryto programos fragmento apvalkalo programos
- apvalkalo programos skaičiuojamos reikšmės.

Apvalkalo programoje apibrėžiami vartotojo parengto fragmento tikrinimui reikalingi pradiniai duomenys ir organizuojamas patikrinimo rezultato išsaugojimas tipiniame rezultatų faile. Šis apvalkalas testuojamajam pateikiamoje užduotyje nerodomas.

## **4.2. Kompiuterizuotos testavimo sistemos struktūra**

Programavimo įgūdžių testavimo sistemos kūrimui pasirinkau jau esamą KTU elektroninio mokymo-testavimo sistemą TestTool (šiuo metu realizuojama 4 versija). Ši mano sprendimą lėmė:



- kad tai nuotolinio testavimo sistema, kurią sudaro:
  - Autoriaus programa - skirta kurti klausimų variantus. Sukurti variantai išsaugomi į failą, kurį vėliau galima įkelti į TestTool4 serverį. Iš įkeltų variantų administratorius formuoja klausimus ir testus.
  - Administratoriaus programa - skirta TestTool4 sistemos administravimui. Per interneto naršyklės sąsają administratorius gali tvarkyti sistemos vartotojus ir jų grupes, įkelti klausimų variantus, formuoti klausimus ir testus, nustatyti egzaminus bei peržiūrėti testavimo rezultatus ir statistiką.
  - Studento programa - skirta testams spręsti. Tai Java kalba parašyta programa, pateikianti studentui pasirinkto egzamino klausimus. Be to studento programa turi du režimus: praktikos ir kontrolės.
- klausimų tipų įvairovė; (gali būti 4 tipų: vieno ir kelių atsakymų išrinkimo; grafinis klausimas; grafinių komponentų manipuliacija (algoritmų modeliavimas); programos teksto analizė);
- sistema sukurta panaudojant šiuolaikines programavimo technologijas, mokymo objektai saugomi XML failuose, o tai reiškia kad ateityje tokią sistemą bus galima suderinti su EMS standartais, palaikančiais XML technologiją. Be to jau yra sudarytis šios testavimo sistemos standartizavimo modelis.

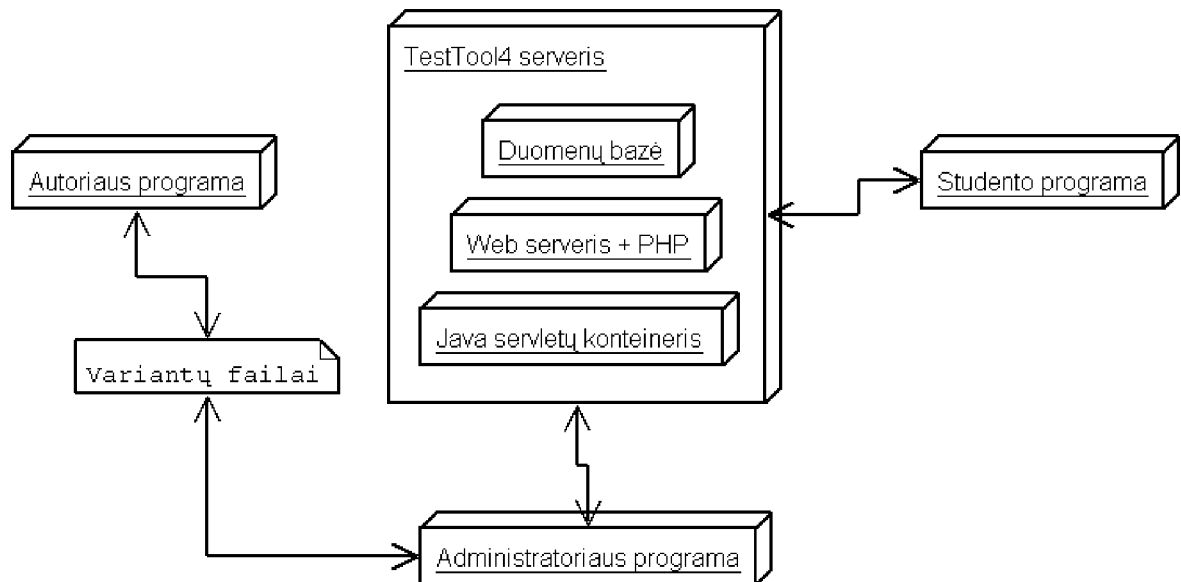
Tačiau programavimo įgūdžiams testuoti šių privalumų nepakanka (sk.4.1.):

- programų tekstų analizės uždavimai atlikti, kai mokinio atsakymai analizuojami grafinio testavimo sistemoje TestTool, programų tekstų perkėlimui į TestTool grafinį redaktorių bus kuriamas konverteris.
- programų taisymo ir rašymo uždavimams realizuoti bus kuriama mokomoji aplinka, kuri studentams pateiktų programavimo uždavimus, griežtus reikalavimus ir programų fragmentus, o studentų sudarytos ar sukomponuotos programos būtų tikrinamos panaudojant programavimo kalbos kompiliatorių.

### **4.3. Testavimo sistemos posisteminių funkciniai modeliai**

Programavimo įgūdžių testavimo sistemos sėkmingas darbas priklauso ir nuo to, kaip gerai tarpusavyje suderinti skirtingi funkciniai modeliai ir kaip efektyviai yra naudojami DB.

Pasiriktos EMS TestTool bendras sistemos veikimo modelis (žr pav 4.1.)



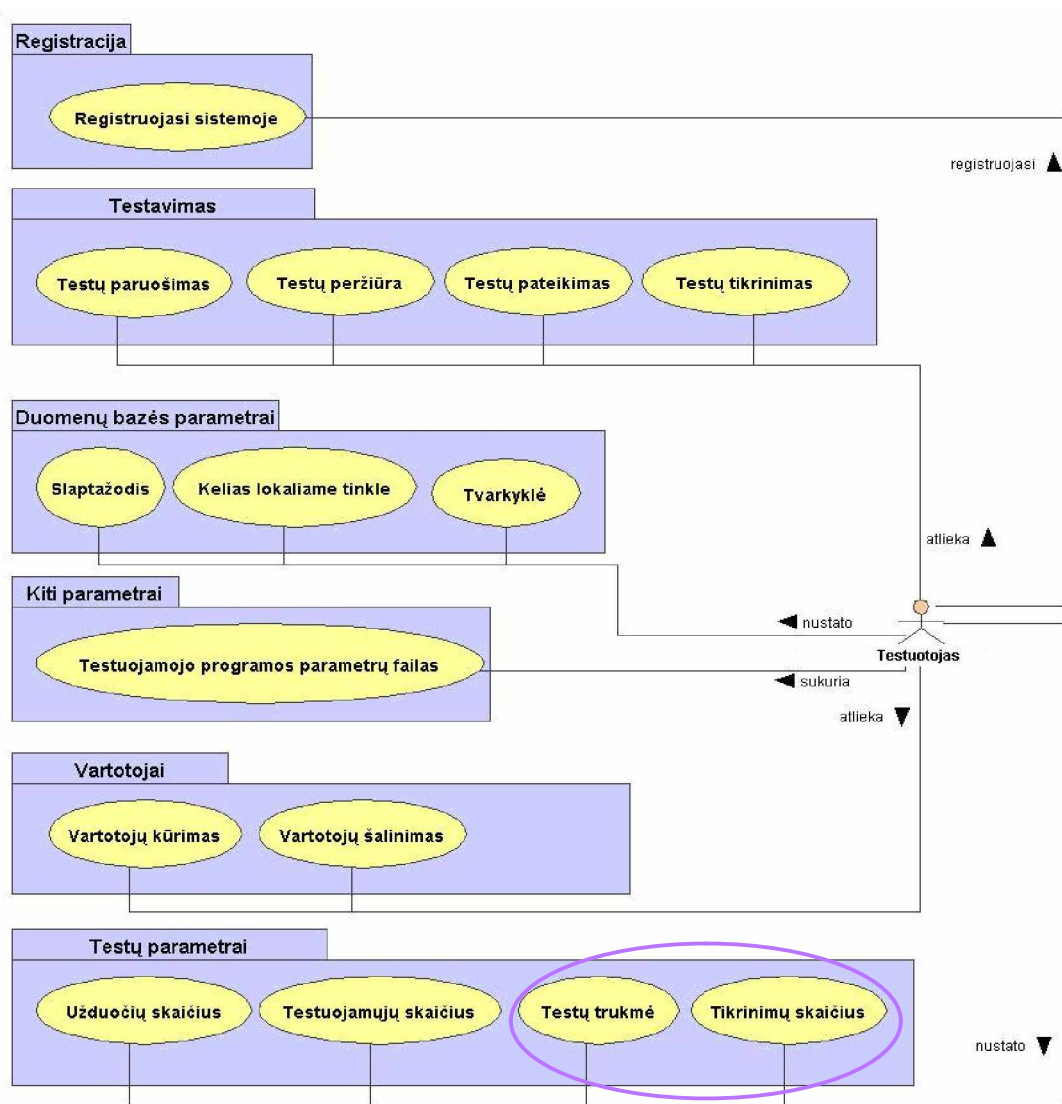
**4.1 pav. Bendras sistemos TestTool4 veikimo modelis**

Autoriaus (dėstytojo) aplinkoje yra realizuotos priemonės testo klausimams paruošti, pateikti bei tikrinti. Šiuo metu dar autoriaus darbui naudojama senesnės versijos (TestTool3) autoriaus programa. Ją sukurti variantų failai yra nesuderinami su naująja TestTool4 sistema, todėl, prieš įkeliant į serverį variantus reikia konvertuoti į reikiamą formatą. Tai yra laikinas reikalavimas, nes šiuo metu kuriama nauja autoriaus programa .

Testuojamojo aplinkoje – atitinkamos priemonės pateiktoms užduotims atlikti bei jų atsakymams perduoti testuotojui. Tarpinė jungtis tarp abiejų klientų programų yra bendra duomenų bazė, nagrinėjamoje TestTool v.4 yra realizuota MySQL duomenų bazėje ir saugoma TomCat serveryje. Architektūros požiūriu ji yra Webservice su nutolusiais klientais Studentais, Autoriais ir Dėstytojais (Administratoriais).

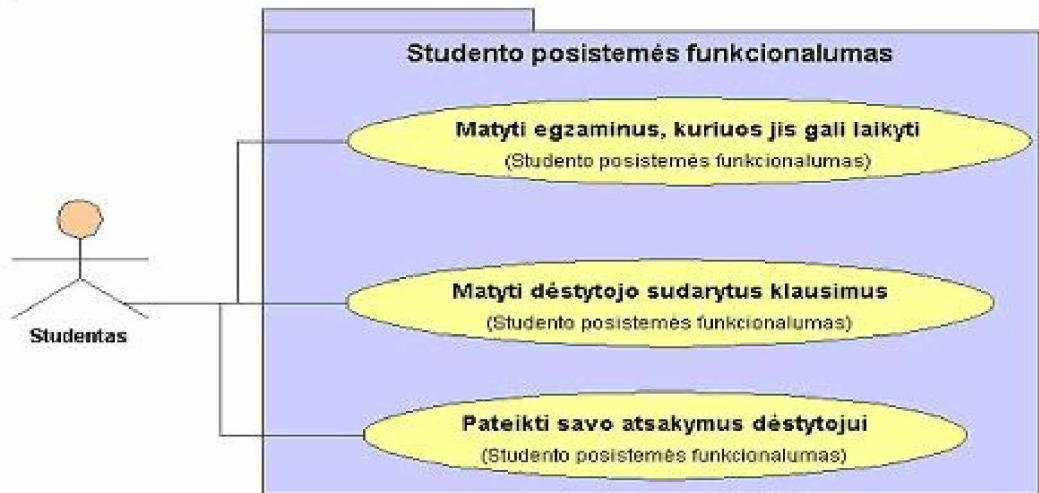
Pateiksiu administratoriaus, dėstytojo bei studentų funkcinius modelius, kai esama sistema bus papildoma naujomis funkcinėmis galimybėmis.

Pritaikius TestTool programavimo įgūdžiams testuoti, esantis dėstytojo funkcinis modelis papildomas naujomis funkcijomis: testų parametruose pildomai reiks nustatyti kiek kartų studentas galės pasinaudoti kompiliatoriumi.



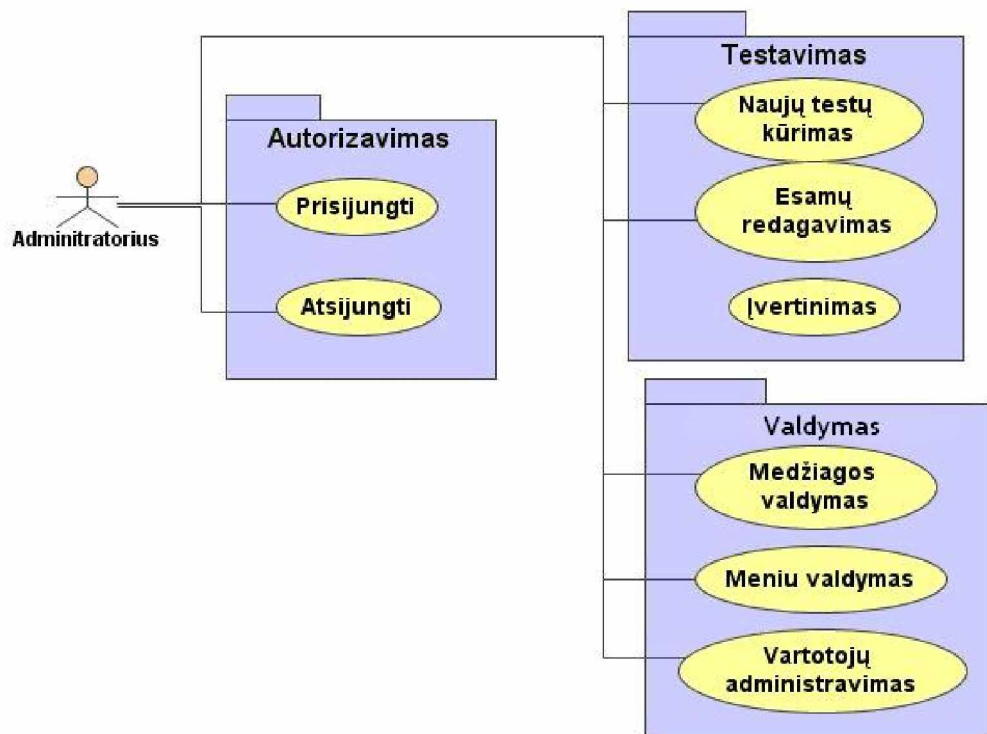
**4.2 pav. Dėstytojo posistemės funkcionalumo schema**

Kinta ir studento darbo galimybės: prisijungimo vardą ir slaptažodį, darbo režimus, leistiną laiką, užduočių kiekį ir įvairovę apibrėžia dėstytojas. Praplėtus sistemos funkcines galimybes, atsižvelgiant į programavimo užduočių tipus (žr. skyrius 4.1.1), studentui atliekant praktinę programavimo užduotį, suteikiama galimybė pasinaudoti kompiliatoriaus paslaugomis. Tačiau bendra funkcionalumo schema lieka nepakitusi:



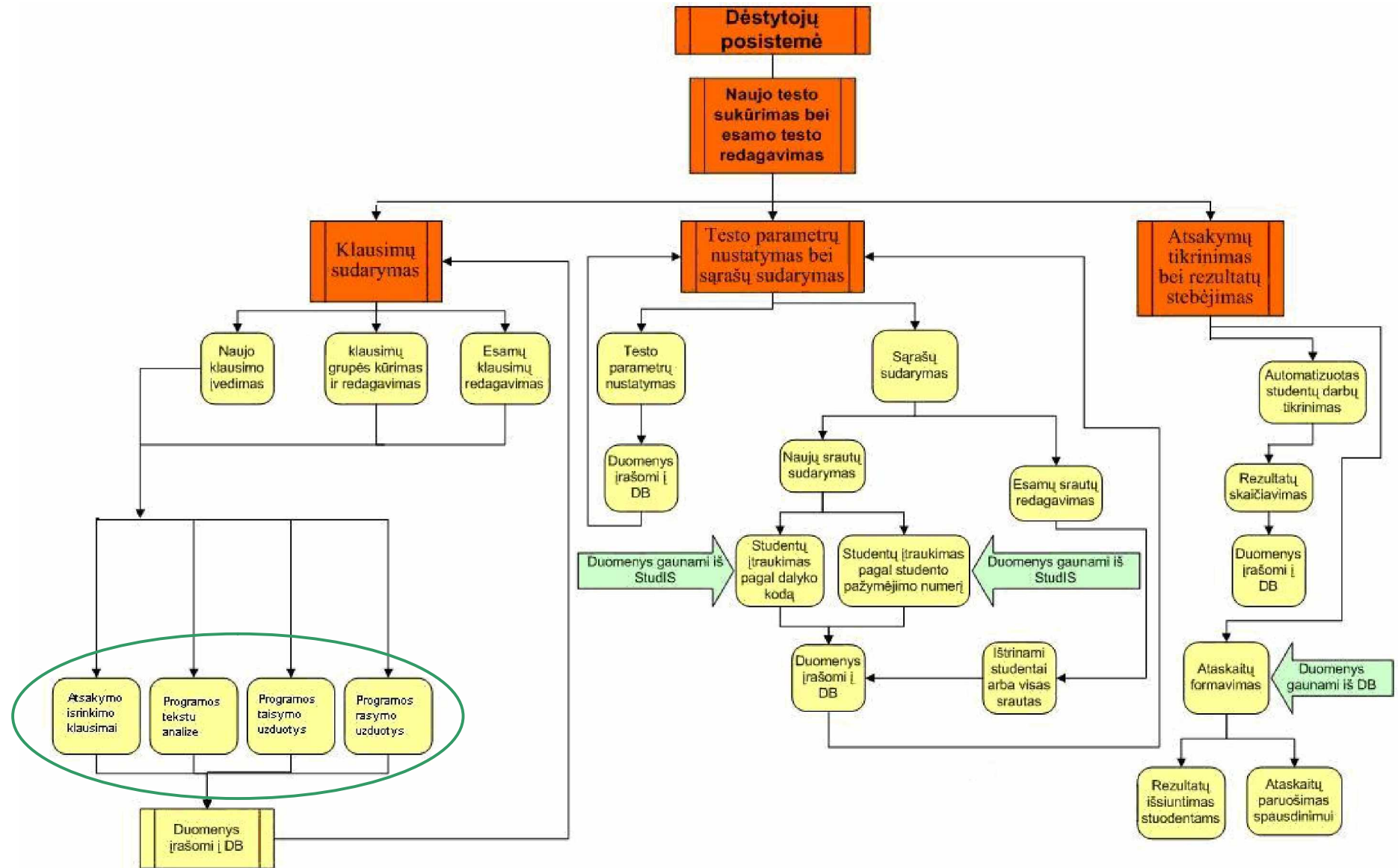
4.3 pav. Studentų posistemės funkcionalumo schema

Administratoriaus kaip ir dėstytojo ir sąsajos veikimo principas iš esmės toks pat, skiriasi funkcijų atliekami veiksmai, administratoriaus funkcionalumas matomas schemoje:



4.4 pav. Administratoriaus posistemės funkcionalumo schema

Išanalizavus autoriaus (dėstytojų) bei studentų funkcijas galima sudaryti jų posistemių modulių veikimo schemas (žr schema)

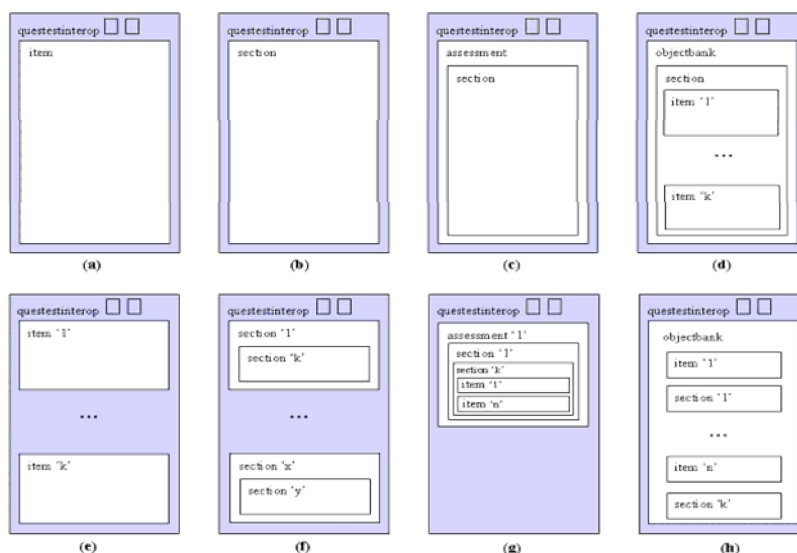


#### 4.4. XML duomenų struktūra klausimams ir testams aprašyti

Testavimo sistemoje klausimams aprašyti naudojama XML duomenų struktūra.

IMS standartų klausimų ir testų suderinamumo (IMS QTI) specifikacija pateikia XML duomenų struktūrą, klausimams ir testams aprašyti. Pagrindinės duomenų struktūros, kuriomis galima keistis, naudojant IMS QTI, pateikiamos žemiau:[8]

- Elementas (Item) – viename QTI-XML faile gali būti vienas arba daugiau elementų. Elementas yra mažiausias nepriklausomas vienetas, kuriuo galima keistis naudojant IMS QTI. Elementas negali būti sudarytas iš kitų elementų. Jis yra daugiau nei klausimas, nes savyje talpina patį klausimą, atvaizdavimo instrukcijas, vartotojo atsakymo apdorojimo instrukcijas, patarimus kartu su teisingu sprendimu bei metaduomenis apibūdinantį elementą [9];



- Skyrius (Section) – viename QTI-XML faile gali būti vienas arba daugiau skyrių. Skyrius savyje gali talpinti tiek kitus skyrius, tiek elementus. Skyrius naudojamas dviem poreikiams tenkinti:
  - Atvaizduoti skirtingas, apibrėžtas tam tikrų mokymo paradigmų grupavimo konstrukcijas[8];
  - Suvaržyti paeiliui sekančių instrukcijų dydį, tam kad būtų valdomi skirtingų galimų sekų sudarymo būdai[8];
- Vertinimas (Assessment) - viename QTI-XML faile gali būti tik vienas vertinimas. Kiekvienas vertinimas būtinai turi turėti bent vieną skyrių, tokiu būdu elementai tiesiogiai negali būti vertinime (nebent kaip skyriaus dalis). Vertinime yra visos reikalingos instrukcijos, kad būtų galima vykdyti elementų kintamųjų trasavimą bei sudėtinis įvertinimas kiekvienam elementui, kad gauti galutinį pažymį[8];

- Objektų bankas (Object bank) – tai tarpusavyje surištų duomenų objektų rinkinys (skyrių ir/arba elementų) [8].

Reikia pažymėti, jog skyriai paprasčiausiai yra skirti grupuoti. Tai leidžia sugrupuoti bet kurio lygio elementus ir skyrius.

Pagrindinės rezultatų pateikimo duomenų struktūros yra:

- Rezultatas – rezultatų aibė tiesiogiai susijusi teisingais įvertinimais. Viename IMS QTI rezultatų pateikimo pakete gali būti keletas atskirų rezultato atvejų vienam dalyviui[8];
- Kontekstas – tai bendra informacija apie pateikiamą rezultatą, t. y. dalyvio vardas ir kiti jo identifikatoriai[8];
- Apibendrintas rezultatas – apibendrinta informacija apie konkretų atskirą įvertinimo atvejį. Kiekvienas rezultatas gali turėti tik vieną apibendrintos informacijos aibę[8];
- Vertinimas-rezultatas – detali informacija apie konkretų vertinimą. Kiekvienas rezultatas gali turėti informaciją tik apie vieną vertinimą[8];
- Skyrius-rezultatas – detali informacija apie užbaigtus arba jau bandytus atlikti skyrius. Kiekvienas rezultatas gali turėti informaciją apie vieną skyrių[8];
- Elementas-rezultatas – detali informacija apie užbaigtus arba bandytus atlikti elementus. Kiekvienas rezultatas gali turėti informaciją apie vieną elementą[8].

Rezultatų pateikimo komponente gali būti rezultatai iš visų vertinimo, skyriaus ir elemento komponentų. Tai reiškia, jog yra paprasta naudoti IMS QTI rezultatų pateikimo XML failą, tam kad pateikti rezultatus, gautus iš įvertinimo, pagrįsto IMS QTI.

Norint pritaikyti šią specifikaciją, reikia naudoti *IMS Question & Test Interoperability: ASI Best Practice & Implementation Guide* [9] dokumentą. Jame pateiktos ir aprašytos visos XML žymės, kurios naudojamos specifikacijoje. Priede pateikiamas teisingo atsakymo išrinkimo XML pavyzdys.

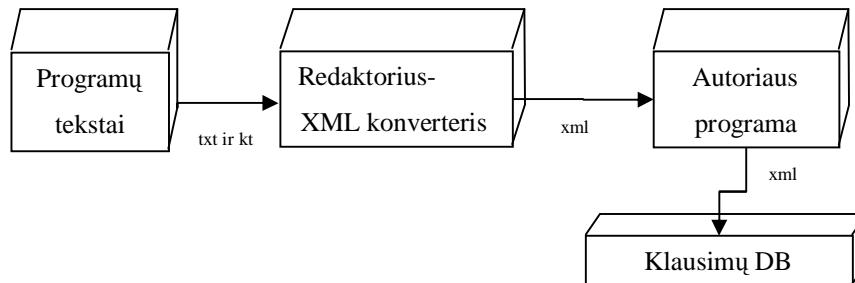
#### **4.5. Programinės realizacijos projektavimas**

Projektui realizuoti bus naudojama prototipo kūrimo metodika. Šis metodas pasirinktas, nes dar kūrimo procese vartotojas gali susipažinti su veikiančia sistema, pamatyti sistemos galimybes, galima aptikti klaidas. O objektinis projektavimo metodas leidžia EMS pritaikyti naujus funkcinius modelius.

#### 4.5.1. Redaktoriaus - XML konverterio sudarymas

Paskirtis: programų tekstų perkėlimui į TestTool grafinį redaktorių, sudėtingiems programų tekstų analizės testams ruošti.

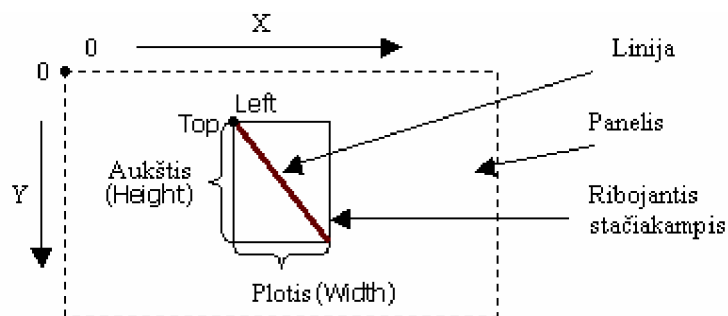
Veikimo modelis:



4.7 pav. Redaktoriaus-XML konverterio veikimo modelis

Pradiniai duomenys – programų tekstai, importuojami iš failų (\*.txt, \*.pas ir kitų programavimo aplinkų). Redaktoriaus pagalba juos galima redaguoti: pašalinti, įterpti, atstatyti pakeitimus ir perkelti į TestTool grafinį redaktorių. (konvertuojama į XML).

Sudarant konverterį, reikia atsižvelgti į šiuos autoriaus programos reikalavimus: grafinius klausimo komponentus (šiuo atveju naudojama **String** – grafinis tekstas) galima išdėstyti tik specialiaame lauke – panelėje. Jo plotis neturi viršyti 450 pikselių, kitaip dalis panelio bus



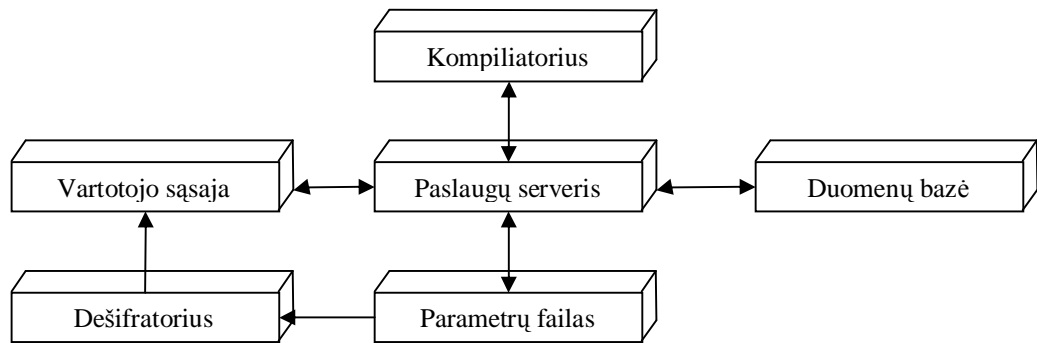
nematoma studento programoje. Kiekvienas grafinis objektas tame tarpe ir (String) turi bazinį tašką, kurio vietą koordinatų pradžios atžvilgiu nusako savybės **Top** ir **Left**

Pilnai paruošti klausimai, įtraukiami į DB.

#### 4.5.2. Kompilatoriaus įdiegimas

Bet kuri testavimo sistema turi būti parengiama taip, kad būtų lengvai modifikuojama ir pritaikoma pasikeitusiems vartotojų poreikiams. Tokius reikalavimus atitinka 10 paveikslėlyje parodyta struktūra, į kurią, keičiant joje įkomponuotą kompiliatorių, nesunku pritaikyti įvairioms programavimo kalboms.[11]

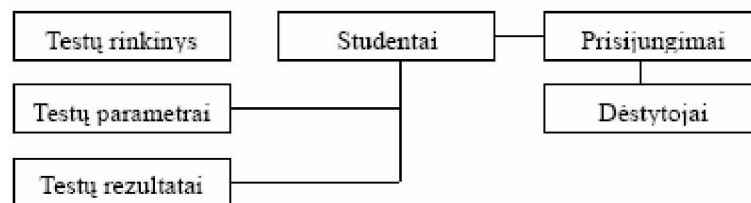




**4.8 pav. Testavimo sistemos struktūra, įkomponavus kompiliatorių**

Parametrų faile laikomi vartotojų prisijungimo kontrolei reikalingi duomenys, informacija apie jiems skiriamų testų tipus ir parametrizuotų duomenų bazės užklausų SQL kalba tekstai. Vartotojo sąsajoje apibrėžiami užklausų parametrai ir jos perduodamos paslaugų serveriui, kuris valdo ryšio su reliacine duomenų baze ir kompiliatoriumi veiksmus. Sistemoje formuojamos trijų grupių vartotojų sąsajos ir jų valdomos paslaugos: sistemos administravimo, autoriaus (dėstytojo) ir studentų.

Reliacinę duomenų bazę sudarančios lentelės ir jų sąryšiai pavaizduoti 4.10 pav. Šioje duomenų bazėje saugomi testų aprašymai, visų testavimų rezultatai ir testavimo sąlygų aprašymai, o taip pat registruojami duomenys apie prisijungusius prie sistemos vartotojus. Aprašant testavimo sąlygas, nurodoma, kiek kurios grupės užduočių reikia atlikti, kiek joms skiriama laiko ir kiek kartų leidžiama kompiliuoti parengtus programų tekstus.



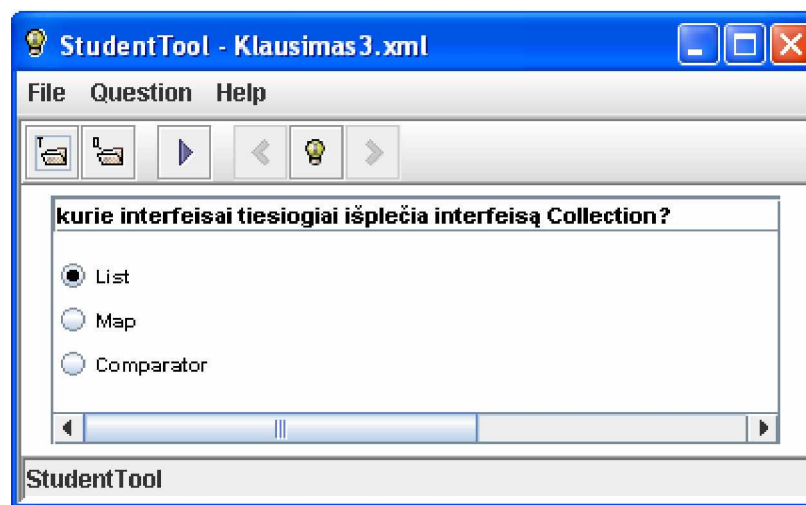
**4.9 pav. Sistemoje naudojamos reliacinės duomenų bazės struktūra**

## 5. PROGRAMAVIMO ĮGŪDŽIŲ TESTAVIMO SISTEMOS EKSPERIMENTINIS TYRIMAS

### 5.1. Užduočių tipų realizavimas

Kaip jau buvo minėta, vienas iš pagrindinių darbo tikslų - elektroninę mokymo sistemą TestTool pritaikyti programavimo įgūdžiams testuoti. Kad įgyvendinti šį tikslą, ši testavimo sistema turi realizuoti šių tipų testus: apklausos; (vieno ar kelių atsakymų pasirinkimas); programų tekstų analizės, programų taisymo ir programų rašymo. (žr. skyrius 5.1).

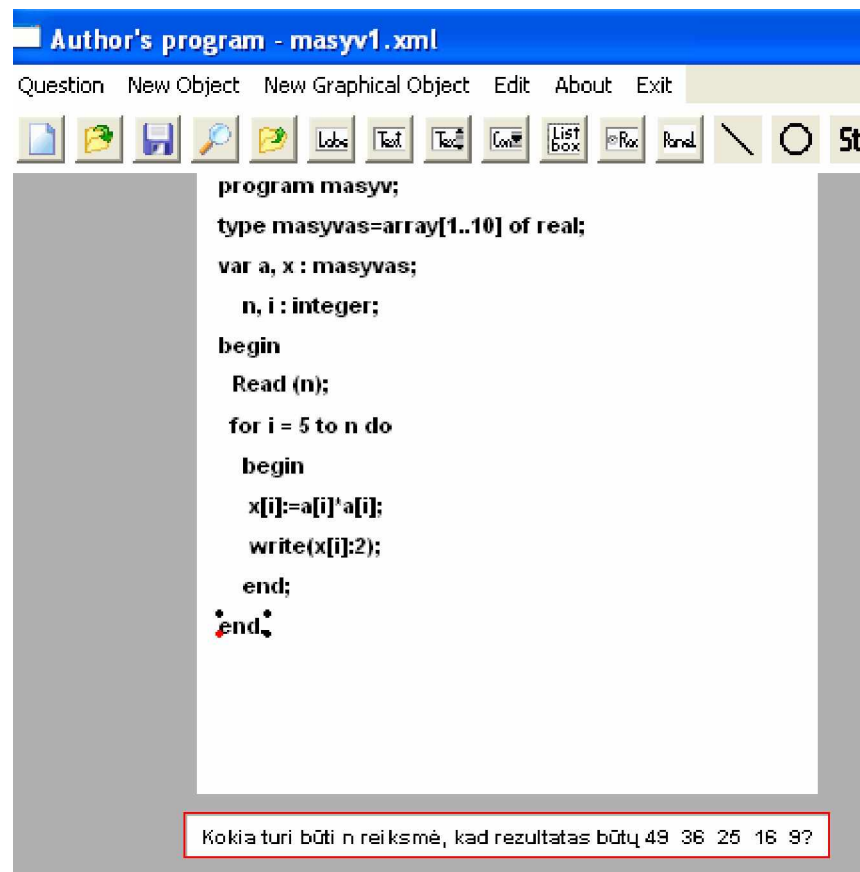
**Vieno ar kelių atsakymo pasirinkimo testai** realizuoti nagrinėjamoje sistemoje Vaizdžiai tai galima parodyti pavyzdžiu iš testavimo sistemos TestTool:



5.1 pav Vieno ar kelių atsakymų pasirinkimo testas

Šio tipo užduotys yra lengviausiai realizuojamos kompiuteriu. Testuojamajam pateikiamos užduočių sąlygos su galimais atsakymų variantais. Teisingas atsakymas ne būtinai gali būti vienas. Paprastas ir pastarojo tipo užduočių vertinimas – tereikia sulygtinti atsakymus, kuriuos pateikė testuojamasis su tais, kurie iš tikrųjų yra teisingi. Išlieka gana didelė tikimybė, kad teisingas atsakymas gali būti atspėjamas.

**Programų tekstų analizės klausimai**, skirti programos teksto analizei, koregavimui, papildymui arba konstravimui. Šiuo atveju reikia pabrėžti, kad atsakymo įvertinimo tikslumas labai priklauso nuo užduočių sudarinėtojo. Jis kaip galima teisingą atsakymą turėtų nurodyti tokį teksto fragmentą, kuris kiek galima labiau atitiktų pilną atsakymą į klausimą. Tokiu būdu sudarinėjant užduotis, gali susidaryti situacija, kai testavimo sistema neteisingą atsakymą užskaitytų kaip teisingą. To išvengti galima maksimaliai sukonkretinant užduotis. Pavyzdžiui, jei užduotis reikalauja aprašyti masyvo tipo kintamąjį, atsakymas bus lengviau įvertinamas, kai užduotyje nurodoma, koks turėtų būti masyvo kintamojo vardas, koks masyvo elementų tipas ir pan.

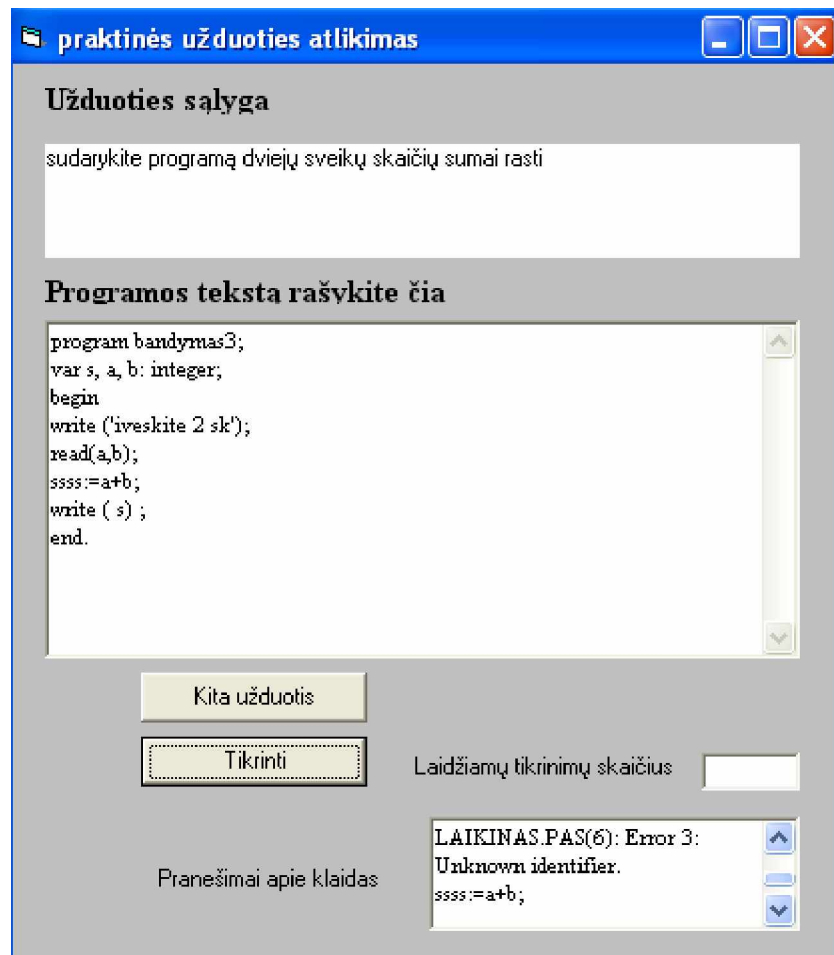


5.2 pav Klausimų analizės testas

**Programos tekstų rašymo tipo klausimai.** Studentas atlikdamas praktinę užduotį gali pasinaudoti kompiliatoriaus paslaugomis tiek taisydamas tiek rašydamas programas (skyrius 4.2.). Sukurtame prototipe yra realizuotos praktinių užduočių, reikalaujančių parašyti programą TPascal kalba, atlikimo priemonės. Pateikiu eksperimentinį šios realizacijos įgyvendinimą:

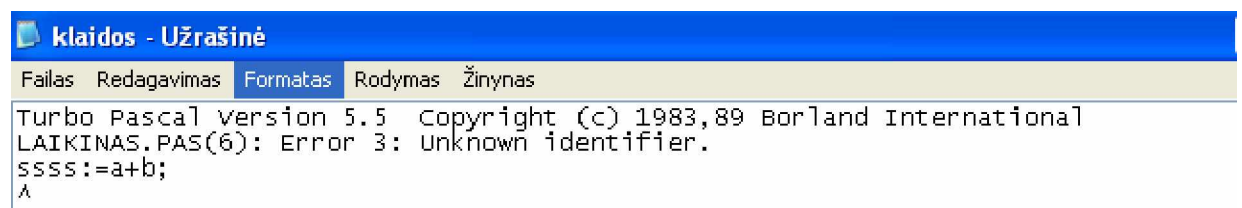
Tokių testų užduotys sudaromos iš keturių dalių:

- pačios užduoties aprašymo,
- komentarų bei papildomų nurodymų testuojamojo rašomam programos fragmentui,
- sudaryto programos fragmento apvalkalo programos
- apvalkalo programos skaičiuojamos reikšmės.



5.3 pav. Kompiliatoriaus panaudojimas praktinei užduočiai atlikti

Sukuriami paketiniai failai, startuojantys kompiliatorių. Pastarasis tikrina parašytą programos kodą bei surašo pranešimus apie klaidas jame į atskirą tekstinį failą:



5.4 pav. Pranešimų apie klaidas tekstinis failas

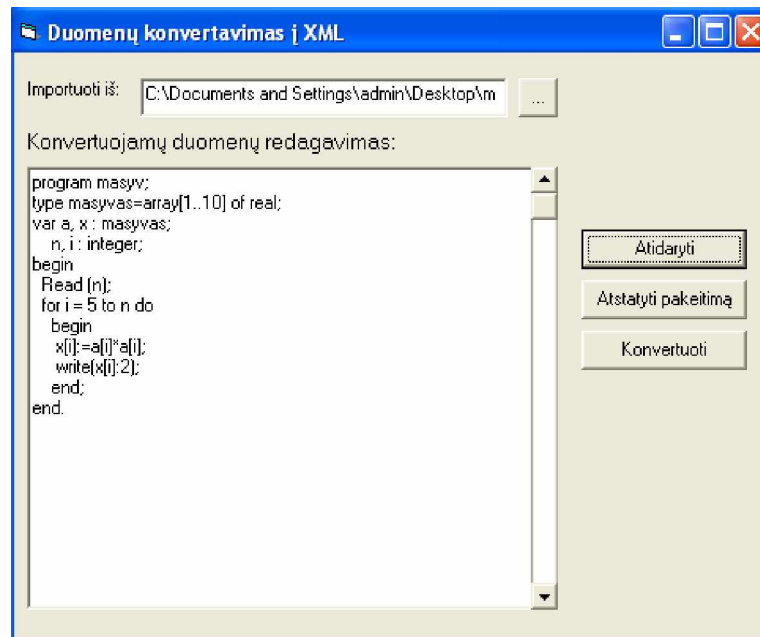
nustato, kuriose eilutėse yra klaidų, kiek jų iš viso yra programoje bei atvaizduoja juos vartotojui (pranešimai apie klaidas). Čia realizuota tik sintaksės klaidų tikrinimas.

Uždavinio sprendimo teisingumui patikrinti su kontroliniais duomenimis reikia sudaryti apvalkalą su tikrinimui reikalingais pradiniais duomenimis.

### 5.1.1. Eksperimentinis redaktoriaus-XML konverterio panaudojimas

Programų tekstų perkėlimui į TestTool grafinį redaktorių, sudėtingiems programų tekstų analizės testams ruošti sukurta priemonė. Pateikiu sukurtos priemonės eksperimentinį pritaikymą.

Pasirenkame importuojamą dokumentą: tai gali būti pavyzdžiui bet kurioje programavimo aplinkoje sudarytas dokumentas. Nuspaudus <Atidaryti>, tekstas esantis byloje perkeliamas į redagavimo sritį



5.5 pav. Duomenų redagavimo ir konvertavimo įrankis

Redagavimo lange galima atlikti pakeitimus: trinti, įrašyti, atstatyti pakeitimus. Eilučių ilgis ir skaičius ribotas, nes turi tilpti į Testtool grafinės aplinkos panelį (žr. sk).

Paspaudus mygtuką <konvertuoti> sukuriamas XML dokumentas. Programų tekstai perkeliama į TestTool grafinį redaktorių. Toliau su autoriaus programa galima dar juos papildyti reikiomomis komponentėmis, keisti jų savybes, tokias kaip spalva, šriftas ir pan.

```

program masyv;
type masyvas=array[1..10] of real;
var a, x : masyvas;
    n, i : integer;
begin
  Read (n);
  for i = 5 to n do
  begin
    x[i]:=a[i]*a[i];
    write(x[i]:2);
  end;
end.*

```

Kokia turi būti n reikšmė, kad rezultatas būtų 49 36 25 16 9?

```

program masyv;
type masyvas=array[1..10] of real;
var a, x : masyvas;
    n, i : integer;
begin
  Read (n);
  for i = 5 to n do
  begin
    x[i]=a[i]*a[i];
    write(x[i]:2);
  end;
end.

```

Kokia turi būti n reikšmė, kad rezultatas būtų 49 36 25 16 9?

### 5.6 pav. Konvertuotas programos tekstas autoriaus ir studento aplinkoje

Taigi tekstų analizės tipo klausimus galima sudaryti žymiai paprasčiau ir greičiau, kadangi nereikia „rankiniu“ būdu perrašinėti jau esamų tekstų.

## 5.2. Atskirų procedūrų bei funkcijų testavimas

Testavimas atliktas tokiais tikslais:

- atrasti ir ištaisyti klaidas, atsiradusias kuriant priemones;
- įrodyti, kad sukurtos priemonės atitinka funkcinis reikalavimus.

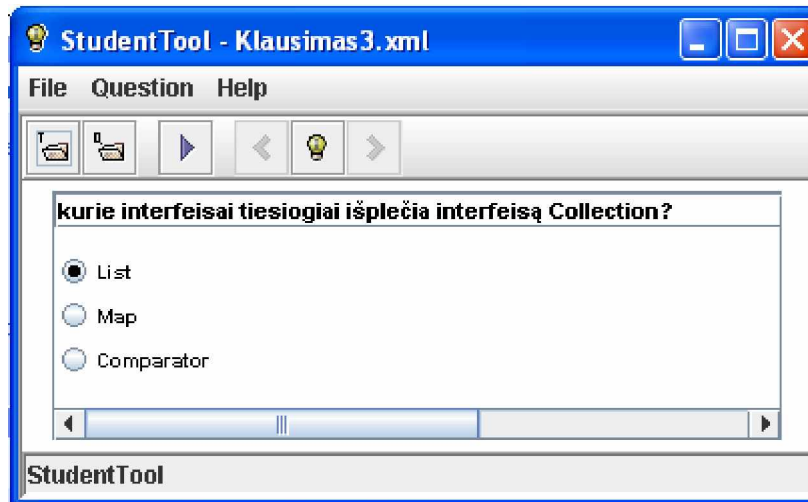
Tam, kad aptikti arba išsiaiškinti ar sukurtos priemonės turi klaidų, reikia ištestuoti jų procedūras bei funkcijas. Kuriant priemones, svarbu užtikrinti jų nepriekaištingą veikimą sistemoje. Daugelis funkcijų ir procedūrų buvo testuojamos kūrimo metu iš karto jas suprogramavus. Testuojant ypatingas dėmesys buvo skiriamas tų funkcijų ribiniams veikimo atvejams: kai įvedami duomenys netenkina įvedamiems duomenims keliamų reikalavimų, kai duomenys išvis neįvedami. Kiekviena funkcija buvo testuojama tol, kol nebūdavo peržiūrėti visi įmanomi ribiniai veikimo atvejai. Jei tam tikroje funkcijoje ar procedūroje buvo randamos klaidos, buvo testuojama tol, kol ištaisoma klaida.

### 5.3. Eksperimentinis testų standartizavimas IMS QTI specifikacija

Kadangi sistema TestTool nors ir nestandartinė, bet turi paruoštą standartizavimo projektą, todėl eksperimento metu pritaikiau standartams keletą klausimų, kurių sudarymui buvo naudojama XML duomenų konverteriu.

IMS QTI specifikacija pateikia XML duomenų struktūrą, klausimams ir testams aprašyti.(nagrinėta 4.2 skyriuje). Sukurti testo klausimų XML failai validuojami naudojant *ims\_qtiasiv1p2p1.dtd* failą.

Pateikiu vieno iš tokių klausimų pavyzdį:



Šis klausimas TestTool sistemoje yra sukurtas kaip XML failas. Šio failo pavyzdys pateikiamas žemiau:

```
<q name="Klausimas3">
- <ob name="Edit" id="obj1">
  - <Uzasas>
    - <![CDATA[
      kurie interfeisai tiesiogiai išplečia interfeisą Collection?
    ]]>
    </Uzasas>
  - <TrText>
    - <![CDATA[
      Text
    ]]>
    </TrText>
  <Top>0</Top>
  <Left>0</Left>
  <Width>55</Width>
  <Fontas>Arial,1,11</Fontas>
  <fSpalva>0</fSpalva>
  <bSpalva>16777215</bSpalva>
  </ob>
- <ob name="RadioGroup" id="obj2">
  <VisSelltem>0</VisSelltem>
  <TrText>0</TrText>
  <Top>1</Top>
```

```

<Left>0</Left>
<Pilnas>>false</Pilnas>
<Fontas>System,1,10</Fontas>
<fSpalva>0</fSpalva>
<bSpalva>16777215</bSpalva>
- <Nariai>
- <![CDATA[
  List,Map,Comparator,
]]>
</Nariai>
</ob>
</q>

```

Pagal IMS QTI standartą perdarius šį klausimą, jį aprašantis XML failas *quest1.xml* atrodo sekančiai:

```

<!DOCTYPE questestinterop SYSTEM "ims_qtiasiv1p2p1.dtd">
<questestinterop>
  <qticomment>Tai testas, kuriame reikia pasirinkti teisinga atsakymą.</qticomment>
  <item title="Teisingos atsakymo pasirinkimas" ident="klaus_3_programavimas">
    <presentation label="klausimas3">
      <flow>
        <material>
          <mattext> kurie interfeisai tiesiogiai išplečia interfeisą Collection? ?</mattext>
        </material>
        <response_lid ident="klausimas3" rcardinality="Single" rtiming="No">
          <render_choice shuffle="Yes">
            <flow_label>
              <response_label ident="A">
                <material>
                  <mattext>List</mattext>
                </material>
              </response_label>
            </flow_label>
            <flow_label>
              <response_label ident="B">
                <material>
                  <mattext>Map</mattext>
                </material>
              </response_label>
            </flow_label>
            <flow_label>
              <response_label ident="C">
                <material>
                  <mattext> Comparator </mattext>

```



```

        </material>
      </response_label>
    </flow_label>
      </render_choice>
    </response_lid>
  </flow>
</presentation>
<resprocessing>
  <outcomes>
    <decvar vartype="Integer" defaultval="1"/>
  </outcomes>
  <rescondition>
    <qticomment>Teisingas variantas.</qticomment>
    <conditionvar>
      <varequal respident="klausimas3">A</varequal>
    </conditionvar>
    <setvar action="Add" varname="SCORE1"> 10 </setvar>
    <displayfeedback feedbacktype="Response" linkrefid="Correct"/>
  </rescondition>
</resprocessing>
<itemfeedback ident="Correct" view="Candidate">
  <flow_mat>
    <material>
      <mattext>Jus atsakete teisingai.</mattext>
    </material>
  </flow_mat>
</itemfeedback>
</item>
</questestinterop>

```

## IŠVADOS

1. Analizės dalyje, apžvelgus elektroninio mokymo sistemas internete, galima teigti, kad negali egzistuoti testavimo sistema, apimanti visas elektroninio mokymosi priemones ir įrankius, kurie gali būti naudojami specializuotoje srityje. Todėl svarbi sistemų savybė - lankstumas arba integracijos galimybė, o ateityje gali išlikti ir būti naudojama tik ta programinė įranga, kuri atitiks kursų aprašymo standartus ir bus suderinta su kitomis elektroninio mokymo aplinkomis.
2. Atlikus analogiškų sistemų analizę buvo nustatyti kompiuterizuotai programavimo įgūdžių testavimo sistemai keliami reikalavimai.
3. Remiantis analize, programavimo įgūdžiams testuoti pasirinkta EMS TestTool ir :
  - a. Programų tekstų analizės užduočiai atlikti, kai mokinio atsakymai analizuojami grafinio testavimo sistemoje TestTool, programų tekstų perkėlimui į TestTool grafinį redaktorių sukurtas konverteris.
  - b. Programų taisymo ir rašymo užduotims realizuoti sukurta mokomoji aplinka, kurioje studentams pateikiama praktinė programavimo užduotis ir sudaroma galimybė panaudoti programavimo kalbos kompiliatoriumi.(Naudojamas TPascal kompiliatorius)
4. Aptarti preliminarūs testų vertinimo būdai parodo, kad vien kompiuterio vertinimais visiškai pasikliauti negalima. Kiek jie gali būti teisingi, priklauso ir nuo užduočių sudarinėtojo.
5. Šiame darbe, išnagrinėjus elektroninio mokymo standartus, pateikiama XML duomenų struktūra klausimams ir testams aprašyti.
6. Eksperimento metu standartams buvo pritaikyta keletas klausimų.

## LITERATŪROS ŠARAŠAS

1. WebCT virtuali mokymo terpė. Prieiga per internetą <http://www.webct.com> [žiūrėta 2003-12-18]
2. Testavimo posistemė eTest. Prieiga per internetą <http://www.jellybeansoftware.com> [žiūrėta 2003-12-18]
3. Livelink for Learning Management nuotolinio mokymo sistema. Prieiga per internetą <http://www.opentext.com> [žiūrėta 2003-12-12]
4. Colleen nuotolinio mokymo sistema. Prieiga per internetą <http://mail.edu.sigma.se> [žiūrėta 2003-12-18]
5. Olimpiados užduočių perdavimo ir vertinimo sistema. Prieiga per internetą <http://ims.mii.lt> [žiūrėta 2004-04-22]
6. D. Hillmann. Using Dublin Core, Prieiga per internetą <http://purl.org/dc/documents/usageguide>, [žiūrėta 2003-12-18]
7. Juliette, Is SCORM coming out ahead? [2002m Spalio 2d.] [Interaktyvus] [žiūrėta 2003-05-25]. Prieiga per internetą: [http://www.september15.net/log\\_september15\\_archive/000100.html#000100](http://www.september15.net/log_september15_archive/000100.html#000100).
8. IMS Question & Test Interoperability: An Overview, Final Specification Version 1.2. Prieiga per internetą: [žiūrėta 2004-09-25] [http://www.imsproject.org/question/qtiv1p2/imsqti\\_oviewv1p2.html](http://www.imsproject.org/question/qtiv1p2/imsqti_oviewv1p2.html).
9. IMS Question & Test Interoperability: ASI Best Practice & Implementation Guide, Final Specification Version 1.2, [žiūrėta 2004-09-25]. Prieiga per internetą: [http://www.imsproject.org/question/qtiv1p2/imsqti\\_asi\\_bestv1p2.html](http://www.imsproject.org/question/qtiv1p2/imsqti_asi_bestv1p2.html).
10. IMS Global Learning Consortium, [žiūrėta 2004-09-12]. Prieiga per internetą: <http://www.imsglobal.org/>.
11. IMS Content Packaging Information Model, Version 1.1.2 Final Specification , [žiūrėta 2004-09-12]. Prieiga per internetą: [http://www.imsproject.org/content/packaging/cpv1p1p2/imscp\\_infov1p1p2.html](http://www.imsproject.org/content/packaging/cpv1p1p2/imscp_infov1p1p2.html).
12. V .Barzdaitis, A.Mickus, A.Vidžiūnas//Kompiuterinių programavimo įgūdžių testavimo sistemų konstravimo principai: konferencijos pranešimų medžiaga [Kaunas, KTU, 2004]. Kaunas, 2004, p. 21-23.
13. J.Skūpienė//Testavimas informatikos olimpiadose: konferencijos pranešimų medžiaga [Kaunas, KTU, 2004]. Kaunas, 2004, p. 37-41.

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

**EMS** — elektroninio mokymo sistema.

**PVAS** — Programavimo varžybų aptarnavimo sistemos.

**W3C** (The World Wide Consortium) -W3 konsorciumas – organizacija, kuri rūpinasi interneto technologijų kūrimu ir standartizavimu.

**W3C** (World Wide Web Consortium) — organizacija, vystanti HTML standartą.

**SCORM** (*Sharable Content Object Reference Model.*) — Pasaulinis standartas.

**API** — taikomųjų programų programavimo sąsaja.

**IMS QTI** (*IMS question & test interoperability*) —IMS klausimų ir testų suderinamumas: (*interoperability*) galimybė paimti mokymo komponentus sukurtus su vienokiais įrankiais ar platforma ir naudoti tuos pačius komponentus kitoje vietoje su skirtingais įrankiais ar platformomis.

**XML** (eXtensible Markup Language) — WWW dokumentų kūrimo kalba. Atnaujinta ir išplėsta lyginant su HTML.

**Manifestas** — specialusis XML failas.

**SGML** (Standard Generalized Markup Language) — dokumentų aptašymo kalba. Atskiria dokumento turinį ir vaizdavimą. HTML, XML yra šio standarto dalys.

**MySQL-** (My Structured Query Language) labai greitas, daugiavartotojiškas, daugiakanalis, efektyvus SQL DB serveris. Serveris yra skirtas kritiniams, didelių apimčių sistemoms palaikyti. Naudojamas pagal dvi licenzijas: atviro kodo ir komercinę.

## PRIEDAS 1 Standartas teisingo atsakymo pasirinkimo klausimui

Pavyzdys:

[http://www.imsproject.org/question/qtiv1p2/imsqti\\_asi\\_bestv1p2.html#1402293](http://www.imsproject.org/question/qtiv1p2/imsqti_asi_bestv1p2.html#1402293) :

Pateikiamas klausimas

**Which one of the listed standards committees is responsible for developing the token ring specification ?**

- IEEE 802.3
- IEEE 802.5
- IEEE 802.6
- IEEE 802.11
- None of the above.

Teisingo atsakymo išrinkimo XML specifikacija turėtų būti

```

<questestinterop>
1   <qticomment>
2       This is a simple multiple-choice example that conforms to V1.2.
3       The rendering is a standard radio button style.
4       Response processing is incorporated.
5   </qticomment>
6   <item title="Standard Multiple Choice Item"
7   ident="IMS_V01_I_BasicExample002b">
8       <presentation label="BasicExample002a">
9           <flow>
10              <material>
11                  <mattext>
12                      Which one of the listed standards committees is
13 responsible
14                      for developing the token ring specification ?
15                  </mattext>
16              </material>
17              <response_lid ident="MCb_01" rcardinality="Single"
18 rtiming="No">
19                  <render_choice shuffle="Yes">
20                      <response_label ident="A">
21                          <flow_mat>
22                              <material><mattext>IEEE 802.3</mattext></material>
23                          </flow_mat>
24                      </response_label>
25                      <response_label ident="B">
26                          <flow_mat>
27                              <material><mattext>IEEE 802.5</mattext></material>
28                          </flow_mat>
29                      </response_label>
30                      <response_label ident="C">
31                          <flow_mat>
32                              <material><mattext>IEEE 802.6</mattext></material>
33                          </flow_mat>
                      </response_label>

```

```

34         <response_label ident="D">
35             <flow_mat>
36                 <material><mattext>IEEE 802.11</mattext></material>
37             </flow_mat>
38         </response_label>
39         <response_label ident="E" rshuffle="No">
40             <flow_mat>
41                 <material><mattext>None of the above.</mattext>
42             </material>
43         </flow_mat>
44     </response_label>
45 </render_choice>
46 </response_lid>
47 </flow>
48 </presentation>
49 <resprocessing>
50     <outcomes>
51         <decvar vartype="Integer" defaultval="0"/>
52     </outcomes>
53     <respcondition title="Correct">
54         <conditionvar>
55             <varequal respident="MCb_01">A</varequal>
56         </conditionvar>
57         <setvar action="Set">1</setvar>
58         <displayfeedback feedbacktype="Response" linkrefid="Correct"/>
59     </respcondition>
60 </resprocessing>
61 <itemfeedback ident="Correct" view="Candidate">
62     <flow_mat>
63         <material><mattext>Yes, you are right.</mattext></material>
64     </flow_mat>
65 </itemfeedback>
66 </item>
</questestinterop>

```

This XML code is available in the file: 'ims\_qtiasiv1p2/basic/mchc\_ir\_001/mchc\_ir\_001a.xml'.