

Методы организации систем электронного обучения на основе структурно-иерархического подхода

05.13.11 - математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание ученой степени кандидата физико-математических наук

Надежда Сергеевна Силкина

Научный руководитель:
СОКОЛИНСКИЙ Леонид Борисович,
доктор физ.-мат. наук, профессор

Актуальность

1. Повсеместное внедрение электронного обучения
2. Перенос образовательного контента из одной системы электронного обучения в другую
3. Модернизация электронных учебных курсов
4. Адаптация электронных учебных курсов (кастомизированное обучение)
5. Необходимость создания модели электронного обучения, поддерживающей понятие дидактической структуры образовательного контента

Дидактическая структура образовательного контента

- ❑ Теория
- ❑ Вопросы по теории с открытым ответом
- ❑ Методические указания по решения задач
- ❑ Задачи с открытым ответом
- ❑ Вопросы по теории с закрытым ответом
- ❑ Задачи с закрытым ответом
- ❑ Литература

Цель диссертационной работы

Разработка модели, методов
и алгоритмов электронного обучения,
поддерживающих дидактическую
структуру образовательного контента

Основные задачи

1. Разработать структурно-иерархическую дидактическую (СИД) модель электронного обучения
2. Разработать операции СИД модели над электронными энциклопедиями, образовательными программами и курсами
3. Разработать алгоритмы анализа качественных характеристик образовательных программ и электронных учебных курсов с использованием операций СИД модели
4. Разработать программную поддержку СИД модели

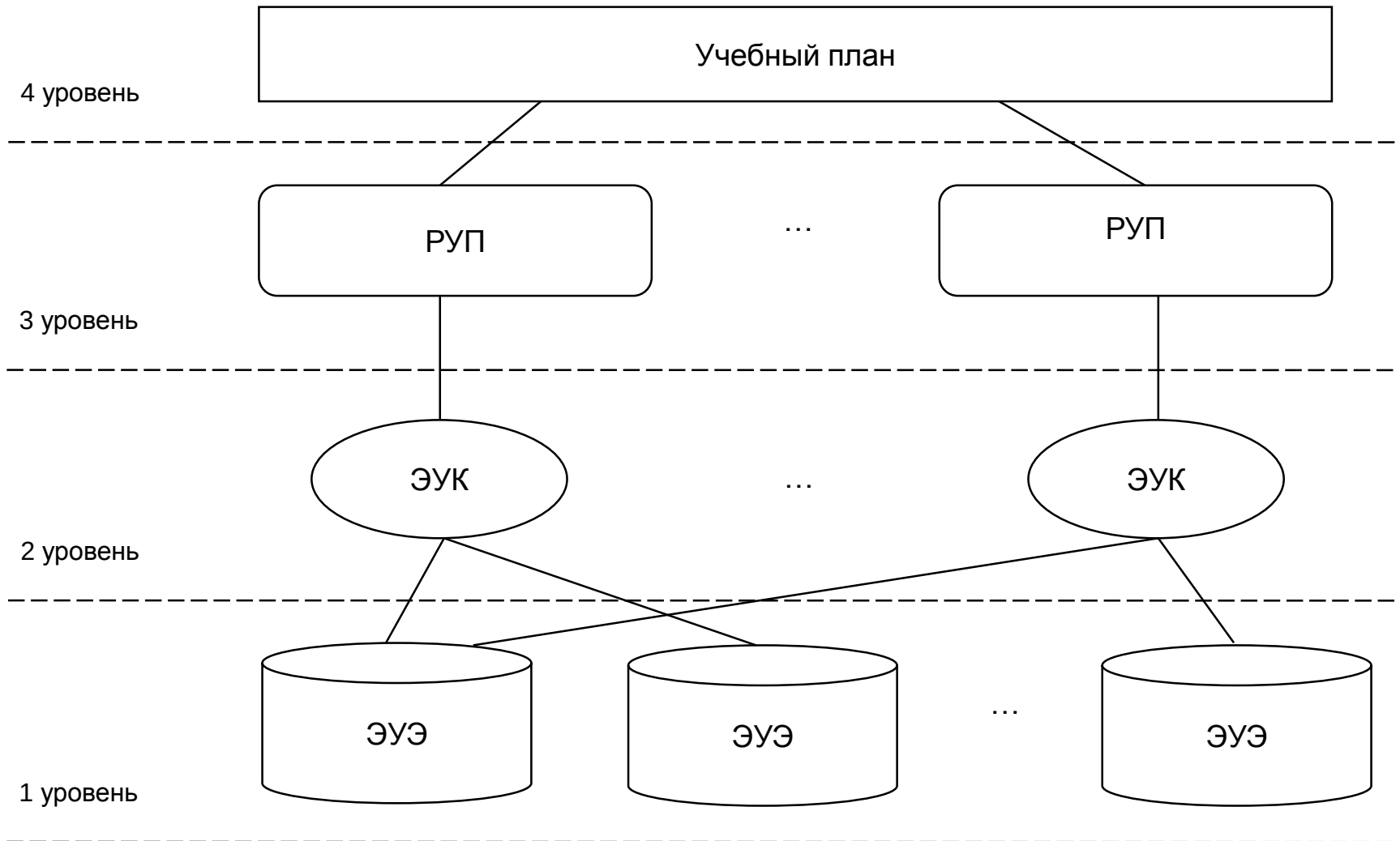
Работы по теме диссертации

<i>Cunningham W., Leuf B.</i> The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Professional, 2001. 464 p.	Электронная энциклопедия
<i>Kristensen T., Lamo Y., Hinna K.R., Hole G.O.</i> Dynamic Content Manager – A New Conceptual Model for E-Learning // Proceedings of the International Conference on Web Information Systems and Mining (WISM '09). Springer-Verlag, Berlin, Heidelberg, 2009. P. 499-507.	Карта понятий
<i>Курганская Г.С.</i> Модель представления знаний и система дифференцированного обучения через Интернет на его основе // Известия Челябинского Научного Центра. 2000. Вып. 2. С. 84-88.	Поток знаний
<i>Соловов А.В.</i> Математические модели содержания и процессов электронного обучения // Телекоммуникации и информатизация образования, 2006, №4. С. 20-37.	Граф содержания
<i>De-Marcos L., Pages C., Martinez J.J., Gutierrez J.A.</i> Competency-Based Learning Object Sequencing Using Particle Swarms // Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence ICTAI 2007 (29-31 Oct. 2007). Vol. 2. P. 111-116.	Компетентностный подход
SCORM 2004 4th Edition. Advanced Distributed Learning (ADL) Initiative. 2009.	Модели электронного обучения базового уровня

Требования к модели

1. Учитывать специфику образовательных стандартов ФГОС 3++
2. Определять дидактические типы для образовательных объектов
3. Поддерживать модульный подход при формировании образовательного контента
4. Предусматривать возможность создания дидактически структурированных хранилищ образовательных объектов в определенной предметной области (электронных энциклопедий)
5. Предусматривать возможность создания дидактически структурированных электронных курсов, а также обеспечивать возможность их автоматизированного создания на основе электронных энциклопедий и других электронных курсов
6. Обеспечивать совместимость с существующими стандартами в области электронного обучения

Структурно-иерархическая дидактическая модель



Дидактические типы

1. **CONCEPT**: теоретическое (текстовое) описание понятия;
2. **OPEN_QUESTION**: вопрос с открытым ответом для самопроверки понимания теоретического описания понятия;
3. **EXAMPLE**: пример решения задачи с использованием изучаемого понятия;
4. **EXERCISE**: задание, проверяющее умение применять изученное понятие при решении задач;
5. **CLOSE_QUESTION**: вопрос с закрытым ответом для проверки освоения теоретического описания понятия;
6. **PROBLEM**: практическое задание, формирующее навыки использования изученного понятия при решении задач;
7. **BIBITEM**: библиографическая ссылка.

Формальное определение базовых элементов

Компонента

$$w = \langle id_{component}, T, \xi, \lambda \rangle,$$

$id_{component}$ — идентификатор компоненты,
 T — дидактический тип компоненты,
 $\xi \in \mathcal{D}_T$ — коллекция, представляющая компоненту,
 λ — трудоемкость компоненты.

Модуль

$$\mu = \langle id_{module}, \vec{w} \rangle,$$

id_{module} — уникальный идентификатор модуля,
 $\vec{w}$ — вектор компонент модуля.

Энциклопедия

$$R = \langle id_{encyclopedia}, M \rangle,$$

$id_{encyclopedia}$ — уникальный идентификатор энциклопедии,
 M — коллекция модулей энциклопедии.

Компетенции в СИД модели

$$K = \langle source, type, number, description \rangle,$$

source — уникальный идентификатор источника, содержащий перечень компетенций,

type — тип компетенции с возможными значениями из множества {УК, ОПК, ПК},

number — порядковый номер компетенции,

description — наименование компетенции.

$$\mathbb{K} = \mathbb{K}_{univer} \cup \mathbb{K}_{genprof} \cup \mathbb{K}_{prof},$$

$$\mathbb{K}_{univer} \cap \mathbb{K}_{genprof} = \emptyset,$$

$$\mathbb{K}_{univer} \cap \mathbb{K}_{prof} = \emptyset,$$

$$\mathbb{K}_{genprof} \cap \mathbb{K}_{prof} = \emptyset.$$

$\mathbb{K}$ — множество всех возможных компетенций,

$\mathbb{K}_{univer}$ — подмножество универсальных компетенций,

$\mathbb{K}_{genprof}$ — подмножество общепрофессиональных компетенций,

$\mathbb{K}_{prof}$ — подмножество профессиональных компетенций.

Стандарты и образовательные программы в СИД модели

$$S = \langle id_{standart}, K_{ungen}, \beta_{stud}, \beta_{pract}, \beta_{crt-l}, \beta_{crt-h}, \beta_{total} \rangle,$$

$id_{standart}$ — идентификатор стандарта;

$K_{ungen} \subset (\mathbb{K}_{univer} \cup \mathbb{K}_{genprof})$ — множество универсальных и общепрофессиональных компетенций;

β_{stud} — минимальное количество зачетных единиц, отводимых на освоение учебных дисциплин;

β_{pract} — минимальное количество зачетных единиц, отводимых на практики;

β_{crt-l} — нижняя граница для зачетных единиц, отводимых на ГИА;

β_{crt-h} — верхняя граница для зачетных единиц, отводимых на ГИА;

β_{total} — общее количество зачетных единиц.

$$Q = \langle id_{syllabus}, id_{standart}, K_{prof}, C \rangle,$$

$id_{syllabus}$ — уникальный идентификатор образовательной программы,

$id_{standart}$ — идентификатор образовательного стандарта,

K_{prof} — множество профессиональных компетенций, предусмотренных образовательной программой,

C — множество курсов, входящих в образовательную программу.

Курс в СИД модели

$$\gamma = \langle id_{course}, id_{syllabus}, \beta_{lec}, \beta_{lab}, \beta_{out}, R, G, f \rangle,$$

id_{course} — уникальный идентификатор курса,

$id_{syllabus}$ — идентификатор образовательной программы,

β_{lec} — трудоемкость лекций,

β_{lab} — трудоемкость лабораторных (практических) занятий,

β_{out} — трудоемкость самостоятельной работы студента,

R — энциклопедия курса,

$G = \langle V, E \rangle$ — граф-план курса,

$f : V \rightarrow R$ — однозначное инъективное отображение, сопоставляющее каждой вершине $v \in V$ граф-плана G некоторый модуль $\mu \in R$: $f(v) = \mu$.

Граф-план в СИД модели

$$G = \langle V, E \rangle$$

$V = \{v_l | l = 1, \dots, b\}$ — множество всех вершин

$E = \{e_p | p = 1, \dots, d\}$ — множество всех ребер



Операции СИД модели

- ❑ операции над энциклопедиями
- ❑ операции над стандартами
- ❑ операции над образовательными программами
- ❑ операции над курсами и граф-планами

Операции над энциклопедиями

Создание новой энциклопедии	CREATE_ENCYCLOPEDIA()
Доступ к атрибутам энциклопедии	GET_ENCYCLOPEDIA_TITLE (encyclopedia) GET_ENCYCLOPEDIA_ANNOTATION (encyclopedia)
Изменение значения атрибута энциклопедии	SET_ENCYCLOPEDIA_TITLE (encyclopedia, value) SET_ENCYCLOPEDIA_ANNOTATION (encyclopedia, value)
Перемещение курсора, указывающего на текущий модуль	RESET_MODULE (encyclopedia) NEXT_MODULE (encyclopedia) PRIOR_MODULE (encyclopedia)
Доступ к текущему модулю	FETCH_MODULE (encyclopedia)
Добавление нового модуля	INSERT_MODULE (encyclopedia)
Удаление текущего модуля	DELETE_MODULE (encyclopedia)
Копирование содержимого модуля module1 в module2	COPY_MODULE (module1, module2)
Доступ к коллекции элементов компоненты <COMPONENT> модуля module	<COMPONENT>_COLLECTION (module) <COMPONENT>: CONCEPT, OPEN_QUESTION, EXAMPLE, EXERCISE, CLOSE_QUESTION, PROBLEM, BIBITEM
Трудоемкость компоненты <COMPONENT> модуля module	<COMPONENT>_CREDIT (module)

Операции над стандартами

Создание нового образовательного стандарта	CREATE_STANDART()
Доступ к атрибутам стандарта	GET_STANDART_<atr> (standart), где <atr>: DESCRIPTION; MIN_STUD, MIN_PRACT, MIN_CRT, MAX_CRT, TOTAL, URL, LEVEL, LENGTH
Изменение значения атрибута стандарта	SET_STANDART_<atr> (standart, value) , где <atr>: DESCRIPTION; MIN_STUD, MIN_PRACT, MIN_CRT, MAX_CRT, TOTAL, URL, LEVEL, LENGTH
Перемещение курсора универсальной компетенции	RESET_UC (standart). NEXT_UC (standart) PRIOR_UC (standart)
Доступ к атрибуту универсальной компетенции	UC_GET_<atr> (standart), где <atr>: NUMBER , DESCRIPTION
Изменение значения атрибута универсальной компетенции	UC_SET_<atr> (standart, value)
Добавление в стандарт новой универсальной компетенции	INSERT_UC_INTO_STANDART (standart, number, description)
Удаление универсальной компетенции из стандарта	DELETE_UC_FROM_STANDART (standart)

Операции над образовательными программами

Создание нового образовательной программы	CREATE_EDU_PROG (standart)
Получение указателя на стандарт	GET_STANDART (edu_prog)
Операции для работы с профессиональными компетенциями	PC_GET_<atr> (edu_prog), PC_SET_<atr> (edu_prog, value), RESET_PC (edu_prog), NEXT_PC (edu_prog), PRIOR_PC (edu_prog), INSERT_PC_INTO_EDU_PROG (edu_prog, number, description), DELETE_PC_FROM_EDU_PROG (edu_prog)
Перемещение курсора списка учебных курсов	RESET_COURSE (edu_prog) NEXT_COURSE (edu_prog) PRIOR_COURSE (edu_prog)
Получение указателя на текущий курс	FETCH_COURSE (edu_prog)
Добавление нового курса	INSERT_COURSE (edu_prog)
Удаление текущего курса	DELETE_COURSE (edu_prog)
Доступ к атрибуту текущего курса	GET_COURSE_<atr> (edu_prog), где <atr>: TITLE, LECTURE, LABORATORY, OUTWORK
Изменение значения атрибута текущего курса	SET_COURSE_<atr> (edu_prog, value), где <atr>: TITLE, LECTURE, LABORATORY, OUTWORK

Операции над курсами и узлами граф-планов

Получение указателя на энциклопедию курса	GET_ENCYCLOPEDIA (course)
Доступ к корню граф-плана курса	GRAPH_PLAN (course)
Перемещение курсора дочерних вершин узла node	RESET_CHILD (node) NEXT_CHILD (node) PRIOR_CHILD (node)
Получение указателя на текущую дочернюю вершину узла node	FETCH_CHILD (node)
Добавление новой дочерней вершины узла node	INSERT_CHILD (node)
Удаление дочерней вершины узла node, на которую указывает курсор	DELETE_CHILD (node)
Связывание узла граф-плана node с модулем module	SET_LINK (node, module)
Добавление номера number в коллекцию номеров универсальных компетенций узла граф-плана node	INSERT_UC_INTO_GP (node, number)
Удаление номера number из коллекции номеров универсальных компетенций узла граф-плана node	DELETE_UC_FROM_GP (node, number)
Определение наличия универсальной компетенции с номером number в узле граф-плана node	UC_ISEXIST (node, number)

Алгоритмы анализа образовательного контента

1. Верификация целостности курса
2. Оценка трудоемкости курса
3. Оценка сбалансированности курса
4. Соответствие образовательной программы стандарту
5. Целостность образовательной программы

Верификация целостности курса

Электронный учебный курс называется *целостным*, если каждый узел граф-плана связан с некоторым модулем

$$R_C = \frac{|V_{linked}|}{|V|},$$

R_C — коэффициент целостности курса,

V_{linked} — множество связанных вершин граф-плана,

V — множество вершин граф-плана.

Оценка трудоемкости курса

Дидактический слой – совокупность всех компонент курса, имеющих одинаковый дидактический тип

Трудоемкость курса – сумма трудоемкостей всех дидактических слоев курса

Трудоёмкость слоя «Теория»

function Subtree_concept_credit(root)

```
1)      module = GET_LINK(root);
2)      credit = CONCEPT_CREDIT(module);
3)      RESET_CHILD(root);
4)      child = FETCH_CHILD(root);
5)      while (child ≠ Null)
6)          credit += Subtree_concept_credit(child);
7)          NEXT_CHILD (root);
8)          child = FETCH_CHILD(root);
9)      end while;
10)     return credit;
end function;
```

function Course_concept_credit(course)

```
1)      root = GRAPH_PLAN(course);
2)      credit = Subtree_concept_credit(root);
3)      return credit;
end function;
```

Трудоемкость курса

```
function Course_credit(course)
```

```
1)  credit = Course_concept_credit(course) +  
      Course_open_question_credit(course) +  
      Course_example_credit(course) +  
      Course_exercise_credit(course) +  
      Course_close_question_credit(course) +  
      Course_problem_credit(course) +  
      Course_bibitem_credit(course);
```

```
2)  return credit;
```

```
end function;
```


Оценка сбалансированности курса

Дисбаланс курса – мера разброса трудоемкостей модулей курса

$$\delta = \frac{\sum_{\mu \in f(V)} \left| \lambda_{avg} - \sum_{i=1}^7 \mu \cdot w_i \cdot \lambda \right|}{|V|} \quad (1)$$

$$\lambda_{avg} = \frac{\sum_{\mu \in f(V)} \sum_{i=1}^7 \mu \cdot w_i \cdot \lambda}{|V|} \quad (2)$$

Соответствие образовательной программы стандарту

1. Ограничения на трудоемкость
2. Покрытие универсальных
и общепрофессиональных компетенций

Ограничения на трудоемкость

```
function Credit_constraints(edu_prog)
1)      standart = GET_STANDART(edu_prog);
2)      study = 0;    practice = 0;
3)      RESET_COURSE(edu_prog);
4)      course = FETCH_COURSE(edu_prog);
5)      while (course  $\neq$  Null)
6)          practice += Course_problem_credit(course);
7)          study += Course_concept_credit(course);
8)          study += Course_open_question_credit(course);
9)          study += Course_example_credit(course);
10)         study += Course_exercise_credit(course);
11)         study += Course_close_question_credit(course);
12)         NEXT_COURSE(edu_prog);
13)         course = FETCH_COURSE(edu_prog);
14)     end while;
15)     if ((study  $\geq$  GET_MIN_STUD(standart)) and (practice  $\geq$  GET_MIN_PRACT(standart))
and (study + practice  $\leq$  GET_TOTAL(standart) - GET_MIN_CRT(standart))
and (study + practice  $\geq$  GET_TOTAL(standart) - GET_MAX_CRT(standart))) then
16)         return true;
17)     end if;
18)     return false;
end function;
```

Покрытие универсальных компетенций

```
function Course_UC_credit(root, uc_number)
1)   module = GET_LINK(root);
2)   if (UC_ISEXIST(root, uc_number)) then
3)       credit = Module_credit(module);
4)   else
5)       credit = 0;
6)   end if;
7)   RESET_CHILD(root);
8)   child = FETCH_CHILD(root);
9)   while (child ≠ Null)
10)       credit += Course_UC_credit(child, uc_number);
11)       NEXT_CHILD (root);
12)       child = FETCH_CHILD(root);
13)   end while;
14)   return credit;
end function;
```

Покрытие

универсальных компетенций

function Course_UC_credit(root, uc_number)

```
1)   module = GET_LINK(root);
2)   if (UC_ISEXIST(root, uc_number)) then
3)       credit = Module_credit(module);
4)   else
5)       credit = 0;
6)   end if;
7)   RESET_CHILD(root);
8)   child = FETCH_CHILD(root);
9)   while (child ≠ Null)
10)      credit += Course_UC_credit(child, uc_number);
11)      NEXT_CHILD (root);
12)      child = FETCH_CHILD(root);
13)   end while;
14)   return credit;
end function;
```

function UC_coverage(edu_prog)

```
1)   standart = GET_STANDART(edu_prog);
2)   RESET_UC(standart);
3)   number = UC_GET_NUMBER(standart);
4)   while (number ≠ Null)
5)       credit = 0;
6)       RESET_COURSE(edu_prog);
7)       course = FETCH_COURSE(edu_prog);
8)       while (course ≠ Null)
9)           root = GRAPH_PLAN(course);
10)          credit += Course_UC_credit(root, number);
11)          NEXT_COURSE(edu_prog);
12)          course = FETCH_COURSE(edu_prog);
13)       end while;
14)       if (credit == 0) then
15)           return false;
16)       end if;
17)       NEXT_UC(standart);
18)   end while;
19)   return true;
end function;
```

Соответствие образовательной программы стандарту

```
function Conformance_to_standart(edu_prog)
1)      if (Credit_constraints(edu_prog) == true)
        and (UC_coverage(edu_prog) == true)
        and (GPC_coverage(edu_prog) == true) then
2)          return true;
3)      else
4)          return false;
5)      end if;
end function;
```

Целостность образовательной программы

Образовательная программа называется

целостной, если выполняются следующие условия:

1. Все курсы образовательной программы являются целостными
2. Образовательная программа соответствует стандарту
3. Профессиональные компетенции образовательной программы покрываются курсами образовательной программы

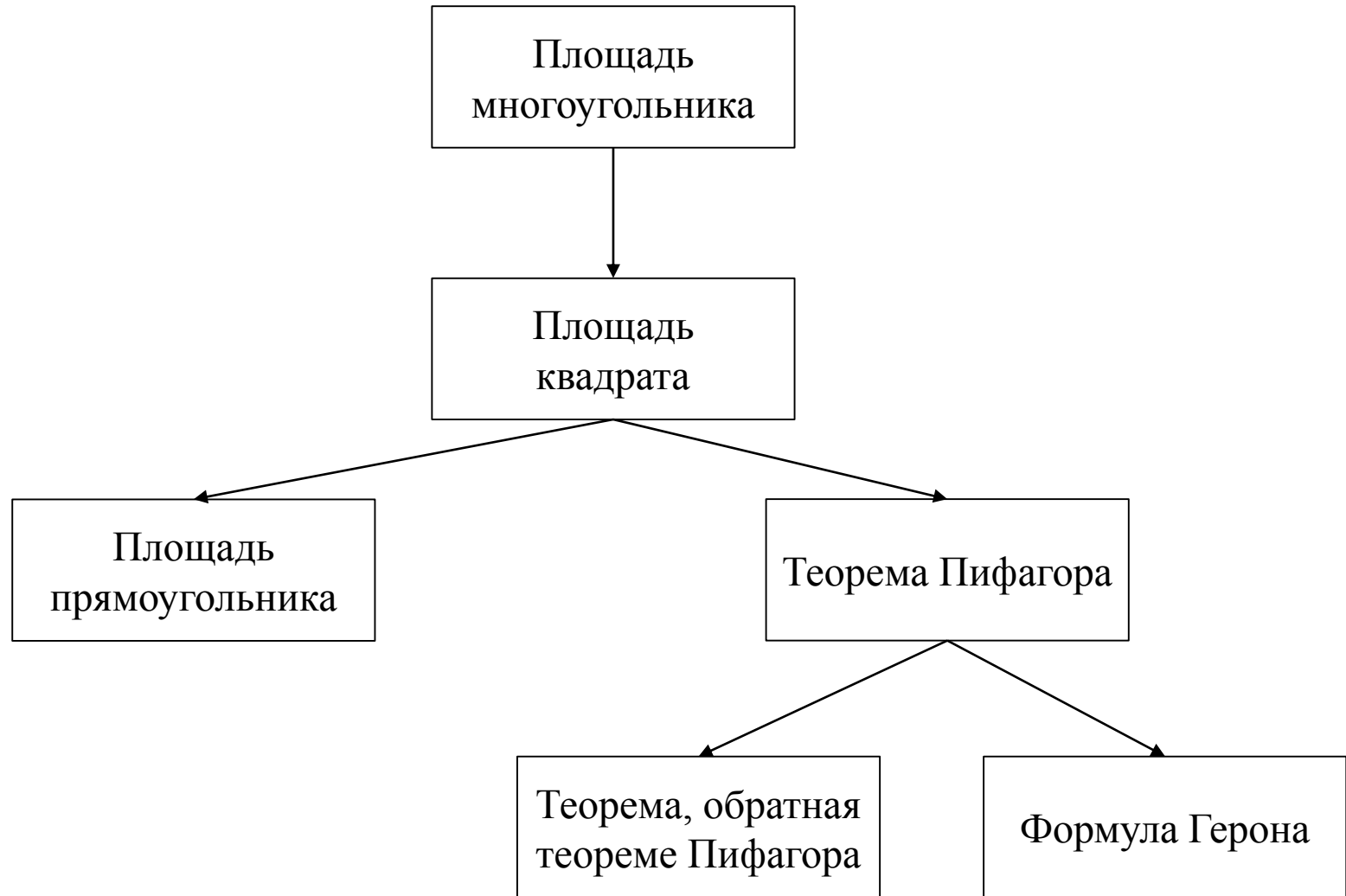
Программная поддержка

- ❑ Разработаны:
 - ❑ Библиотека операций СИД модели
 - ❑ Библиотека алгоритмов анализа образовательного контента
 - ❑ Веб-приложение ECoD, предоставляющее пользовательский интерфейс для манипулирования объектами СИД модели
 - ❑ Функция генерации SCORM-пакета для электронного учебного курса
- ❑ Средства реализации:
 - ❑ Язык программирования PHP
 - ❑ СУБД MySQL
- ❑ Исходные тексты доступны по адресу:
<https://github.com/zhnadya81/ECoD>

Эксперименты

1. Создание миникурса «Площадь многоугольника»
2. Анализ образовательной программы

Создание миникурса



<https://edu.susu.ru/mod/scorm/view.php?id=337468>

Логин: user2019, пароль: Password2019

Создание миникурса

Площадь многоугольника

Меню курса

- ☒ Площадь многоугольника
- ☒ Площадь квадрата
- ☒ Площадь прямоугольника (Зачет 100)
- ☒ Теорема Пифагора
- ☐ Теорема, обратная теореме Пифагора
- ☒ Формула Герона



Теория

Площадь многоугольника

[Теория](#) [Контрольные вопросы](#) [Примеры](#) [Контрольные задания](#) [Тестирование](#) [Литература](#) [Практика](#)

Понятие площади известно из повседневного опыта. Каждый понимает смысл слов: площадь комнаты равна шестнадцати квадратным метрам, площадь садового участка – восьми соткам и т.д.

Можно сказать, что **площадь многоугольника** – это величина той части плоскости, которую занимает многоугольник.

Измерение площадей проводится с помощью выбранной единицы измерения аналогично измерению длин отрезков. За единицу измерения площадей принимают квадрат, сторона которого равна единице измерения отрезков. Так, если за единицу измерения отрезков принят сантиметр, то за единицу измерения площадей принимают квадрат со стороной 1 см. Такой квадрат называется **квадратным сантиметром** и обозначается см^2 . Аналогично определяется **квадратный метр** (м^2), **квадратный миллиметр** (мм^2) и т.д.

При выбранной единице измерения площадей площадь каждого многоугольника выражается положительным числом. Это число показывает, сколько раз единица измерения и ее части укладываются в данном многоугольнике.

На рисунке 1 изображен прямоугольник, в котором квадратный сантиметр укладывается ровно 6 раз. Это означает, что площадь прямоугольника равна 6 см^2 .



Рис. 1.

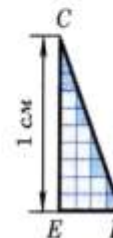
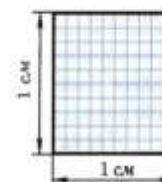
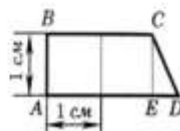
В трапеции ABCD, изображенной на рисунке 2, квадратный сантиметр укладывается два раза и остается часть трапеции – треугольник CDE, в котором квадратный сантиметр не укладывается целиком. Для измерения площади этого треугольника нужно использовать доли квадратного сантиметра, например квадратный миллиметр. Он составляет 0,01 часть квадратного сантиметра. Это показано на рисунке 3, где квадратный сантиметр разбит на 100 квадратных миллиметров (этот рисунок для большей наглядности дан в увеличенном масштабе).

На рисунке 4 видно, что квадратный миллиметр укладывается в треугольнике CDE 14 раз, и остается часть этого треугольника (она закрашена на рисунке голубым цветом), в которой квадратный миллиметр не укладывается целиком. Поэтому можно сказать, что площадь трапеции ABCD приблизительно равна $2,14 \text{ см}^2$.

Оставшуюся часть треугольника CDE можно измерить с помощью более мелкой доли квадратного сантиметра и получить более точное значение площади трапеции.

Описанный процесс измерения можно продолжить далее, однако на практике он неудобен.

Обычно измеряют лишь некоторые связанные с многоугольником отрезки, а затем вычисляют площадь по определенным формулам.



Анализ образовательной программы

1. Создание стандарта

- ❑ Добавлено 8 универсальных компетенций
- ❑ Добавлено 5 общепрофессиональных компетенций

2. Создание энциклопедии

Название модуля	concept	open_question	example	exercise	close_question	problem	bibitem	Трудоемкость модуля
Первый	0,05	0,03	0,04	0,05	0,03	21	0	21
Второй	0,05	0,03	0,04	0,05	0,03	0	0	0.2
Третий	0,02	0,01	0,02	0,03	0,02	0	0	0.1

3. Создание образовательной программы

- ❑ Добавлено 10 профессиональных компетенций
- ❑ Добавлено 56 курсов

Генерация граф-планов

1. Создание корня граф-плана
2. Создание 4 дочерних узлов
 - номер универсальной компетенции : $j + i\%2 * 4$
3. Создание 2 дочерних узлов второго уровня
 - номер общепрофессиональной компетенции : $k + i\%4$
4. Создание 2 дочерних узлов третьего уровня
 - номер профессиональной компетенции : $l + i\%9$
5. Корням граф-планов курсов с порядковыми номерами 1, 2 и 3 ставится в соответствие модуль «Первый», остальным узлам граф-планов ставится в соответствие модуль «Третий»
6. Корням граф-планов остальных курсов ставится в соответствие модуль «Второй», остальным узлам граф-планов ставится в соответствие модуль «Третий»

$j \in [1, 4]$ – номер дочернего узла, $i \in [1, 56]$ – порядковый номер курса, $k \in [1, 2]$ – порядковый номер дочернего узла второго уровня, $l \in [1, 2]$ – порядковый номер дочернего узла третьего уровня

Результаты анализа образовательной программы

№	Критерий	Объект анализа	Результат
1	Коэффициент целостности курса	Все курсы	1
2	Дисбаланс курса	Курсы 1, 2, 3	40.7452
3	Дисбаланс курса	Курсы 4, ..., 56	0.1918
4	Проверка ограничений на трудоемкость	Образовательная программа	True
5	Проверка покрытия универсальных компетенций	Образовательная программа	True
6	Проверка покрытия общепрофессиональных компетенций	Образовательная программа	True
7	Проверка соответствия образовательной программы стандарту	Образовательная программа	True
8	Проверка покрытия профессиональных компетенций	Образовательная программа	True
9	Целостность образовательной программы	Образовательная программа	True

Публикации

Статьи в журналах из перечня ВАК

1. Силкина, Н.С. Система UniCST — универсальная среда электронного обучения / Н.С. Силкина, Л.Б. Соколинский // Системы управления и информационные технологии. — 2010. — № 2. — С. 81–86.
2. Силкина, Н.С. Структурно-иерархическая дидактическая модель электронного обучения / Н.С. Силкина, Л.Б. Соколинский // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». — 2019. — Том 8, № 4. — С. 56-83.
3. Силкина, Н.С. Обзор адаптивных моделей электронного обучения / Н.С. Силкина, Л.Б. Соколинский // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». — 2016. — Том 5, № 4. — С. 61–76.
4. Силкина, Н.С. Модель образовательного стандарта третьего поколения на основе компетентностного подхода для систем электронного обучения / Н.С. Силкина, А.С. Евдокимова // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». — 2011. — № 37(254). — Вып. 10. — С. 90–98.
5. Жигальская (Силкина) Н.С. Моделирование дидактической структуры электронных учебных комплексов // Вестник Южно-Уральского государственного университета. Серия «Математическое моделирование и программирование». 2008. № 27(127). Вып. 2. С. 4–9.

Публикации

Статья в издании, индексируемом в SCOPUS и Web of Science

6. Ivanova, O.N. Competence-Oriented Model of Representation of Educational Content / O.N. Ivanova, N.S. Silkina // Proceedings of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO'2017, Opatija, Croatia, May 22–2

Свидетельство о регистрации программ и баз данных

7. Жигальская (Силкина) Н.С., Евдокимова А.С. Свидетельство Роспатента о государственной регистрации программы для ЭВМ «UnicstLite: программное средство для разработки электронных учебных энциклопедий на базе платформы Microsoft.NET» № 2008614487 от 03.10.2008, правообладатель: ГОУ ВПО "ЮУрГУ".2017. – Р. 791–794.

Основные результаты, выносимые на защиту

1. Разработана структурно-иерархическая дидактическая (СИД) модель электронного обучения
2. Разработаны операции СИД модели над электронными энциклопедиями, образовательными программами и курсами
3. Разработаны алгоритмы анализа качественных характеристик образовательных программ и электронных учебных курсов с использованием операций СИД модели
4. Разработан прототип программной системы ECoD для создания электронных учебных курсов на базе СИД модели