

В. В. Подбельский, О. В. Максименкова, М. А. Казаков

Визуализация структуры педагогического теста

Предложена математическая модель, на базе которой производится визуализация структуры педагогического теста. Разработана программная среда, автоматизирующая построение подобных структур. Описано применение модели для анализа структурных характеристик конкретного теста по дисциплине «Программирование».

Ключевые слова: педагогическое тестирование, педагогический тест, структура теста, тест.

V. V. Podbelsky, O. V. Maksimenkova, M. A. Kazakov

Visualization of a Pedagogical Test Structure

A mathematical model for the educational test structure visualization is considered in this article. Software which is based on this model has been developed. Model application for the analysis of structure characteristics of a certain test on “Programming” academic subject is given in this work.

Keywords: an educational test, a pedagogical test, a test structure, a test.

Введение

При всем многообразии форм промежуточного и итогового контроля знаний в настоящее время широкое распространение в сфере высшего профессионального образования приобретает тестовая форма контроля знаний. Во многом рост интереса к тестированию связан не только с его технологичностью, но и с желанием повысить уровень объективности при оценивании учащихся.

Отметим, что при всех несомненных плюсах и достоинствах существуют неоспоримые недостатки, обусловленные не только формой, но и процедурой тестирования. Особенно ярко некоторые изъяны проявляются в применении тестов при текущем контроле по отдельным дисциплинам. Создание новых тестовых заданий промежуточного, текущего и итогового контроля для формирования базы тестовых заданий по дисциплине – задача трудоемкая, зачастую не включенная в основную нагрузку преподавателей. При этом массовость тестирований по той или иной дисциплине невелика, контингент студентов редко превышает 100–150 человек. Организация повторных тестирований на той же выборке испытуемых практически невозможна [12]. Привлечение экспертов для калибровки заданий и оценивания качества тестов также не всегда возможно ввиду ограниченной численности педагогических кадров, специализирующихся на преподавании конкретных учебных дисциплин вуза.

Для повышения качества тестов в такой ситуации желательно иметь возможность анализа теста на этапе его структурно-содержательного конструирования. Структурно-содержательное конструирование включает этап определения и конкретизации целей тестирования и этап разработки плана и спецификации теста [1, 2, 7, 8]. С точностью и правильностью реализации данных этапов напрямую связаны качественные характеристики теста. Другими словами, на этапе структурно-содержательного конструирования разработчик выбирает, для чего будет применяться тест и что именно он будет измерять. Проверка соответствия теста запланированным целям и его предполагаемому виду обычно требует привлечения экспертов как в области тестологии, так и в предметной области, к которой относится тест.

Этап конструирования теста завершается формированием (конкретизацией) его содержания. Из определения содержания теста (под содержанием теста будем понимать отображение содержания учебной дисциплины в систему тестовых заданий) может быть получена совокупность устойчивых связей между содержанием учебной дисциплины и системой тестовых заданий. Удобным средством визуального представления подобных связей являются графы.

Анализ структуры графа, наглядно представляющего отображение учебной дисциплины на систему тестовых заданий, позволяет выявить проблемы и несоответствия, заложенные в тест еще до его

наполнения конкретными заданиями. Исходя из этого, такой граф будем называть *графом структуры содержания*.

Существуют тесты (например, относящиеся по В. С. Аванесову [1], к нетрадиционным, адаптивным тестам), для которых важна возможность анализа взаимосвязей непосредственно между тестовыми заданиями. Очевидно, что, имея связи тестовых заданий с единицами содержания учебных дисциплин, можно визуализировать связи и между тестовыми заданиями. Графы, отображающие связи между тестовыми заданиями в тесте, будем называть *графами тестовых заданий*.

Однако построение и анализ подобных графов вручную – задача трудоемкая и требующая специальных знаний в области математики и тестологии. Автоматизация процесса позволяет упростить данную процедуру, сократив время на построение и анализ графа, и предоставить:

- разработчику – базу для предварительного анализа теста, минимизирующую необходимость привлечения экспертов;

- эксперту – удобный и наглядный материал, упрощающий и ускоряющий проведение экспертиз.

Постановка задачи визуализации структуры педагогического теста.

Для формализации задачи визуализации необходимо:

- Рассмотреть априорную информацию о педагогическом тесте, аккумулированную в описании его содержания и спецификации теста.

- Получить визуальное представление связей между:

- ✓ тестовыми заданиями и структурными единицами (разделами) учебной дисциплины вне зависимости от вида теста (текущего, промежуточного или итогового контроля);

- ✓ заданиями одного теста;

- ✓ индивидуальными результатами тестирования испытуемых с конкретными единицами учебной дисциплины.

- Ввести обозначения, описать правила и механизмы, связанные с предполагаемой графической моделью.

- Проанализировать возможность оценки предварительного качества тестов на основе полученных графических моделей.

Математическая модель

Спецификация теста является средством представления запланированной содержательной структуры теста и других планируемых автором характеристик [7]. В ней обязательно фиксируется структура, содержание проверки и процентное соотношение заданий в тесте [8]. Из спецификации теста можно сформировать реляционное отношение, связывающее тестовые задания с соответствующими разделами учебной дисциплины (см. табл. 1).

Таблица 1

Соответствие тестовых заданий разделам учебного курса

Тестовое задание	Разделы учебного курса
t_1	D_1
...	...
t_k	D_k
...	...
t_n	D_n

Для формализации введем обозначения:

D – множество разделов учебной дисциплины;

T – множество тестовых заданий, относящихся к D ;

$S \subset D$ – подмножество всех разделов учебной дисциплины, проверяемых T ;

t_i – i -ое задание из T ;

$D_i \subset \mathbb{D} \subset \mathbb{M}$ – подмножество разделов учебной дисциплины, проверяемых t_i ;

D_q – q -ый раздел из \mathbb{M} .

Если t_i проверяет раздел d_k и t_j проверяет раздел d_k , то $D_i \cap D_j = \{d_k\}$.

Для элементов множеств T и \mathbb{D} допустимо построение следующих графов:

- Граф структуры содержания визуализирует связи между конкретными тестовыми заданиями и разделами учебной дисциплины;
- Граф тестовых заданий строится на основе графа структуры содержания и визуализирует связи между тестовыми заданиями через дидактические единицы.

Граф структуры содержания

Правила построения графа структуры содержания:

1. Вершины графа: $t_i \in T$ и $d_k \in \mathbb{D}$; e – ребро графа структуры содержания, $e \in E$, где E – множество ребер графа структуры содержания;

2. $\exists e_{ik} = (t_i, d_k)$ в случае, если задание t_i проверяет раздел d_k ;

3. $\forall i, j \exists e = (t_i, t_j)$;

4. $\forall i, j \exists e = (d_i, d_j)$.

Утверждение 1

Графы структуры содержания всегда двудольные.

Доказательство:

По определению неориентированный граф $G = (W, E)$ называется двудольным, если множество его вершин можно разбить на две части U и V , причем $U \cup V = W$, $|U| > 0$, $|V| > 0$, так, что ни одна вершина из U не соединена с вершинами в U и ни одна вершина из V не соединена с вершинами в V [смотри 6, 9].

Для графа структуры содержания конкретного теста введем обозначения:


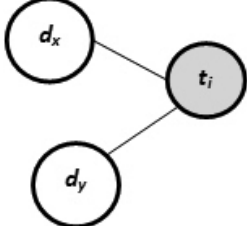
TD – множество вершин графа, TT – множество вершин t : $TT \subset T$, DD – множество вершин d : $DD \subset \mathbb{D}$

Очевидно, что $TD = TT \cup DD$, причем $|TT| > 0$ и $|DD| > 0$.

Из правил построения 3 и 4 видно, что ни одна вершина из TT не соединена с вершинами из TT и ни одна вершина из DD не соединена с вершинами из DD . Следовательно, граф структуры содержания соответствует определению двудольного графа. ■

Правила выделения цветом вершин из множества T в графе структуры содержания

Пусть множество $C = \{c_i\}$, где $0 \leq i \leq 4$ – множество цветов для выделения цветом вершин из множества T . Вершины графов структуры содержания вида будем раскрашивать по следующим правилам, показанным на рис. 1.

$D_i = \{d_x\}, D_i = 1$ вершина t_i окрашивается в цвет c_0 ;	$D_i = \{d_x, d_y\}, D_i = 2$, то вершина t_i окрашивается в цвет c_1 ;
	
$D_i = \{d_x, d_y, d_z\}, D_i = 3$, то вершина t_i окрашивается в цвет c_2 ;	$ D_i \geq 4$, то вершина t_i окрашивается в цвет c_3 ;

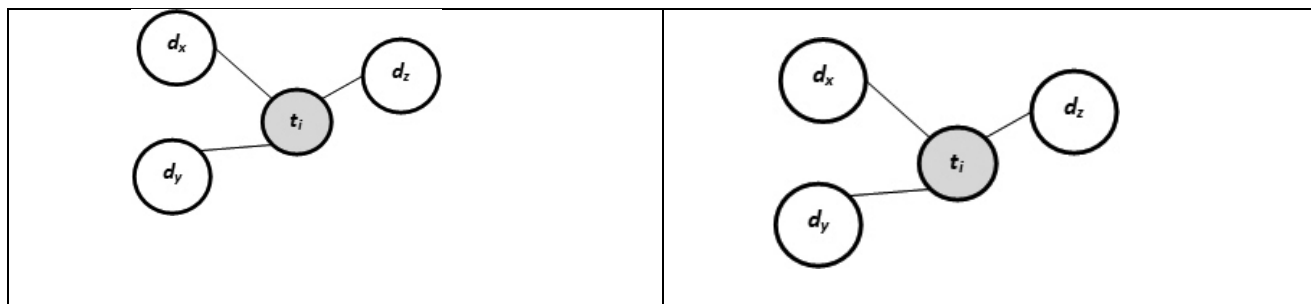


Рис. 1. Цветовое выделение вершин множества Т в графе структуры содержания

Граф тестовых заданий

Правила построения графа тестовых заданий:

1. Вершины графа: t_k ; g – ребро графа тестовых заданий, $g \subseteq G$, где G – множество ребер графа тестовых заданий;

2. $\exists g_{ji} = (t_j, t_i) \Leftrightarrow \exists e_{ki}, e_{kj}: e_{ki} = (d_k, t_i), e_{kj} = (d_k, t_j)$;

3. $\exists g_{ji} = (t_j, t_i) \Leftrightarrow \exists e_{ki}, e_{kj}: e_{ki} = (d_k, t_i), e_{kj} = (d_k, t_j)$.

Пример 1

$|T| = 3; |D| = 2$

$D = \{d_i, d_j\}$

$D_1 = \{d_i\}, D_2 = \{d_i, d_j\}, D_3 = \{d_j\}$

Таблица 2

Пример соответствия тестовых заданий разделам учебного курса

Тестовое задание	Раздел учебной дисциплины
t_1	$\{d_i\}$
t_2	$\{d_i, d_j\}$
t_3	$\{d_j\}$

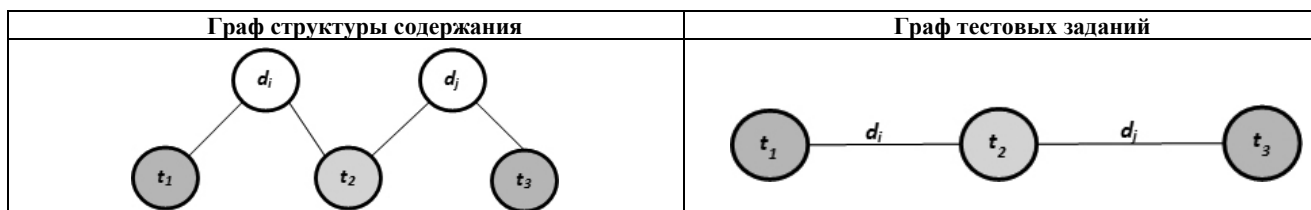


Рис. 2. Пример цветового выделения вершин множества Т в графе структуры содержания

Пример 2

$|T| = 4; |D| = 2$

$D = \{d_i, d_j\}$

$D_1 = \{d_i\}, D_2 = \{d_i, d_j\}, D_3 = \{d_i, d_j\}, D_4 = \{d_j\}$

Таблица 3

Пример соответствия тестовых заданий разделам учебного курса

Тестовое задание	Раздел учебной дисциплины
t_1	$\{d_i\}$
t_2	$\{d_i, d_j\}$

t_3	$\{d_i, d_j\}$
t_4	$\{d_j\}$

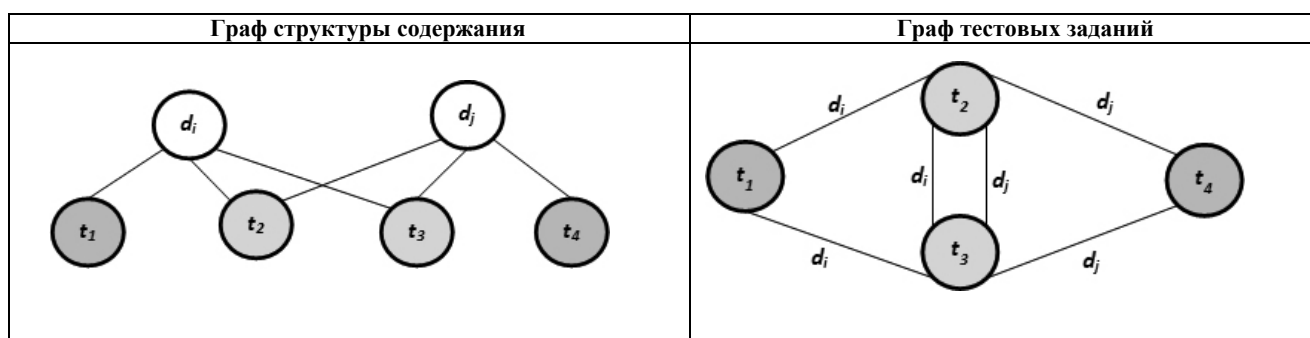


Рис. 3. Пример цветового выделения вершин множества T в графе структуры содержания

Программная реализация

Для автоматизации процесса построения графов математической модели разработана программа с рабочим названием «Программа визуализации структуры педагогического теста».

Для создания программы выбраны следующие технологии:

- Язык C# 4.0. Активно поддерживается и развивается компанией Microsoft, входит в состав Visual Studio 2010.
- MS. NetFramework 4.0. Содержит в себе все необходимые инструменты, которые позволяют эффективно и качественно решить поставленную задачу.
- Windows Presentation Foundation – система для построения клиентских приложений Windows. В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода. Производительность WPF выше, чем у GDI+, за счет использования аппаратного ускорения графики через DirectX.

Программа:

- строит граф структуры содержания и граф тестовых заданий по реляционному представлению спецификации, оформленному в формате документа MS Excel любой версии;
- предоставляет пользователю функционал по управлению как группами, так и отдельными вершинами;
- позволяет добавлять к графу и удалять из графа текстовые и графические метки;
- сохраняет результаты работы в файле графического формата BMP [13, 15].

Основные механизмы работы программы

Поступившая на вход программе спецификация теста преобразуется в коллекцию, каждый элемент которой хранит свое название и названия связанных с ним элементов. Эта коллекция временно хранится в памяти и не видна пользователю.

На основе полученной коллекции строится коллекция пользовательских элементов, соответствующих вершинам, и коллекция пользовательских элементов, соответствующих ребрам.

Для вывода на экран коллекции применяется модифицированный алгоритм поиска в глубину. Сначала на экран выводится какая-либо вершина, затем уровнем ниже на экран выводятся все потомки данной вершины, кроме тех, что уже есть на экране. После чего данная операция применяется к каждому потомку. Если для всех к текущему моменту визуализированных вершин нет потомков, которые еще не выведены на экран, ищется вершина, которой еще нет на экране. Если такая вершина нашлась, то к ней применяется тот же алгоритм. Таким образом, в деревообразной форме выводятся всевозможные компоненты связности. Затем происходит вычисление координат ребер и осуществляется их визуализация на экране.

Пример анализа теста по дисциплине «Программирование»

В национальном исследовательском университете Высшая школа экономики на протяжении более пяти лет на первом курсе бакалавриата специальности 231000–«Программная инженерия» преподается учебный курс «Информатика и программирование». С 2010–2011 гг. курс «Информатика и программирование» получил официальное название «Программирование».

Курс рассчитан на 378 часов: 86 – лекции, 86 – семинары и 204 – самостоятельная работа студентов.

В Высшей школе экономики применяется четырехмодульная система разбиения учебного года. Учебный план и программа курса предусматривают по дисциплине «Программирование» в каждом модуле аудиторные контрольные работы, домашнюю контрольную работу (в третьем модуле); зачет в первом и третьем модулях и экзамены после второго и четвертого модулей. Таким образом, в качестве форм контроля в курсе используются аудиторные контрольные работы, домашние контрольные работы и тесты, применяемые в двух формах: для организации промежуточного и итогового контролей [11, 14, 12].

Рассмотрим пример автоматизированной обработки теста промежуточного контроля, который применялся на зачете по дисциплине «Программирование» за первое полугодие 2010–2011 учебного года. Тест является традиционным гомогенным тестом, охватывает четыре раздела курса и состоит из 30 вопросов. Для гомогенных тестов важно наиболее полное отражение в тестовых заданиях содержания проверяемых разделов учебной дисциплины. Гомогенный тест, таким образом, может содержать незначительное число вопросов, проверяющих две и более темы. Большинство тестовых заданий направлены на проверку материалов одной темы.

Целью данного теста является проверка усвоения учащимися учебного материала по разделам «Т2. Основные элементы программирования (данные, выражения, операторы)», «Т3. Массивы и строки», «Т4. Методы как основа структурного программирования» и «Т5. Класс как контейнер статических методов». На этапе структурно-содержательной проработки в тесте в качестве доминирующей выделена тема 2 «Основные элементы программирования» как содержащая наибольшее количество базовых понятий языка программирования.

Для построения графов структуры содержания и тестовых заданий применимо более мелкое дробление по темам, нежели это предусмотрено программой учебной дисциплины. В табл. 4 выделены темы вопросов (Т2, Т3, Т4, Т5) и приведено их соответствие разделам курса. Для удобства построения графов каждой теме поставлено в соответствие обозначение – маркер темы вопроса ($d_j, 1 \leq j \leq 19$).

Таблица 4

Распределение заданий теста промежуточного контроля по разделам курса «Программирование»

Маркер темы вопроса	Тема вопроса	Раздел курса	Номер вопроса в тесте
d_1	Типы данных	Т2	1
d_1	Типы данных	Т2	2
d_2	Приведение типов	Т2	3
d_2	Приведение типов	Т2	9
d_3	Операции	Т2	4
d_4	Условный оператор	Т2	5
d_5	Универсальный цикл	Т2	6
d_6	Тернарная операция	Т2	7
d_7	Цикл с предусловием	Т2	8
d_8	Оператор ветвления	Т2	10
d_9	Свойства алгоритмов	Т2	11
d_{10}	Цикл с постусловием	Т2	12
d_{11}	break, continue	Т2	12
d_{12}	Одномерный массив	Т3	13
d_{12}	Одномерный массив	Т3	14
d_{13}	Строки	Т3	14
d_{13}	Строки	Т3	17
d_{14}	Многомерный массив	Т3	15
d_{14}	Многомерный массив	Т3	18
d_{14}	Многомерный массив	Т3	19
d_{15}	Форматирование	Т2	16
d_{16}	Методы	Т4	20

d_{16}	Методы	T4	21
d_{16}	Методы	T4	23
d_{16}	Методы	T4	27
d_{17}	Перегрузка методов	T4	22
d_{17}	Перегрузка методов	T4	30
d_{18}	Статические члены класса	T5	24
d_{18}	Статические члены класса	T5	25
d_{18}	Статические члены класса	T5	28
d_{19}	Рекурсия	T4	26
d_{19}	Рекурсия	T4	29

На вход программе подан Excel документ, содержащий информацию только из первого и четвертого столбцов таблицы 1.

Граф структуры содержания промежуточного теста по первому полугодю приведен на рис. 4. Наличие тем, проверяемых одиночными вопросами, для конкретного теста объясняется принадлежностью этих тем к одному, самому крупному разделу учебного курса «Основные элементы программирования», наиболее охваченному вопросами теста (см. табл. 4). Примененное (для наглядности) мелкое дробление разделов учебной дисциплины привело к появлению вершин типа d , связанных с одиночными вершинами типа t .

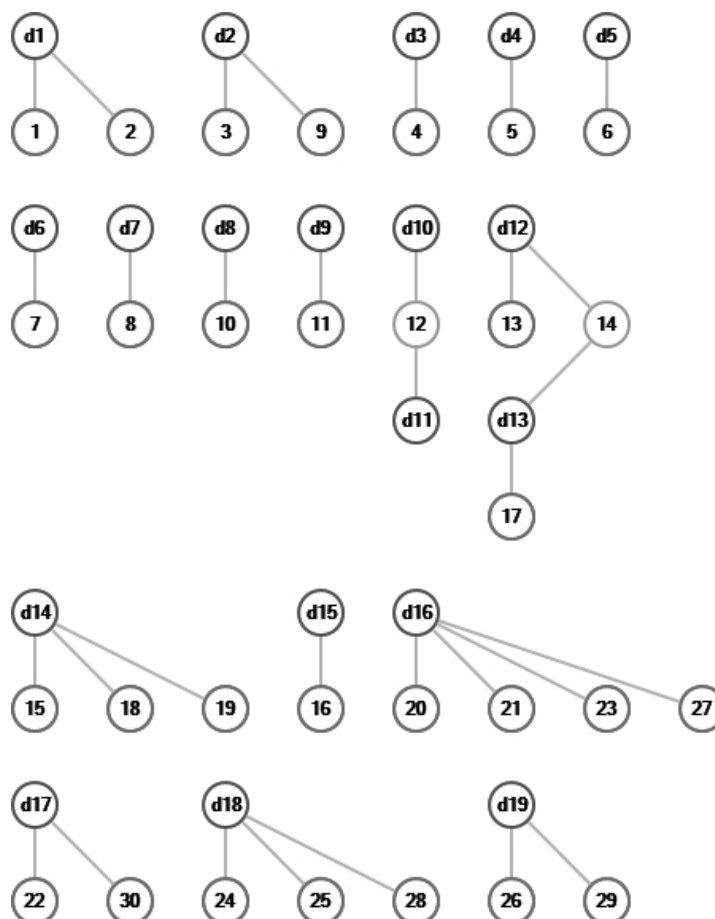


Рис. 4. Граф структуры содержания теста промежуточного контроля по дисциплине «Программирование» (разбиение вопросов по темам)

Используя $T2 = \{d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d15\}$, $T3 = \{d12, d13, d14\}$, $T4 = \{d16, d17, d19\}$ и $T5 = \{d18\}$, визуализируем связь вопросов теста с разделами учебного курса (см. рис. 5).

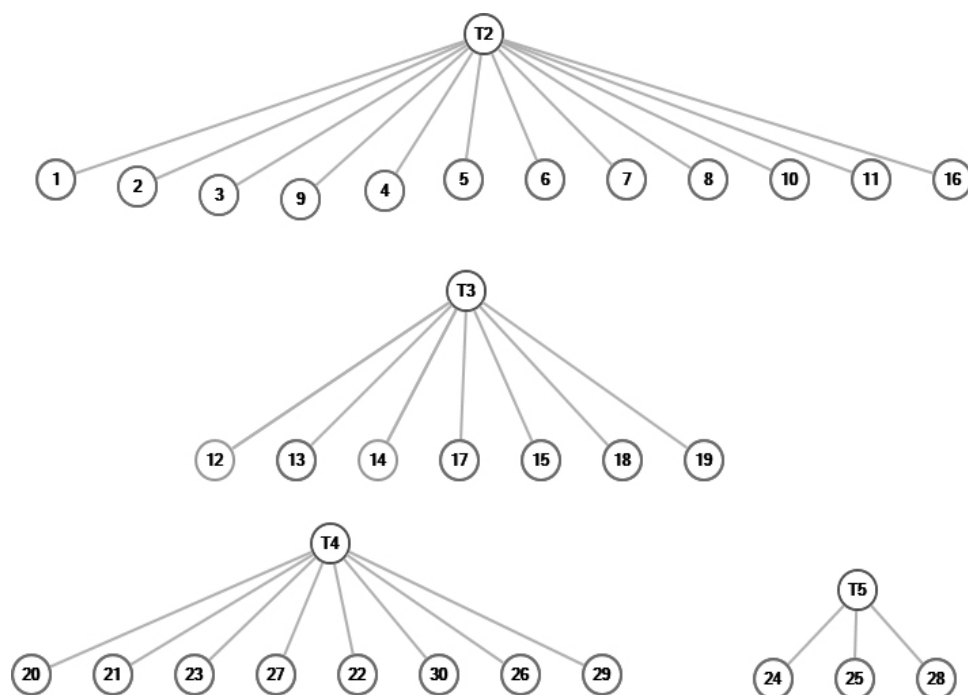


Рис. 5. Граф структуры содержания теста промежуточного контроля по дисциплине «Программирование» (разбиение вопросов по разделам)

Из структуры графов на рис. 4 и 5 видно, что содержание анализируемой учебной дисциплины достаточно полно отражено в исследуемом тесте. Можно сделать вывод, что он вполне применим в качестве зачетного по дисциплине «Программирование». Базовый раздел T2, как видно из рис. 5, представлен в тесте наибольшим числом вопросов, то есть тест вполне соответствует целям, заложенным в него разработчиками. Можно предположить, что данный тест имеет удовлетворительные показатели надежности и валидности, что подтверждается значениями статистических характеристик теста. Подробное описание анализа показателей данного теста и результаты этого анализа приведены в работе [12].

Граф тестовых заданий для данного зачетного теста (теста промежуточного контроля) по первому полугодью представлен на рис. 6.

В присутствующих в графе на рис. 6 компонентах связности наглядно представлены группы вопросов, относящихся к одним и тем же темам курса. Малое количество подобных групп и присутствие множества изолированных одиночных вершин свидетельствует в данном случае о неадаптивной природе рассмотренного теста. В действительности, рассмотренный в качестве примера тест промежуточного контроля по дисциплине «Программирование» не был сконструирован для проведения тестирований в адаптивной форме.

Заключение

Основанная на графах математическая модель, предложенная в работе, позволяет формализовать и частично автоматизировать анализ содержания педагогических тестов. На основе введенного в статье формализма разработана процедура визуализации связей между разделами учебной дисциплины и проверяющими их тестовыми заданиями. Применение этой процедуры визуализации позволяет проводить априорный качественный анализ отображения содержания учебной дисциплины на систему тестовых заданий.

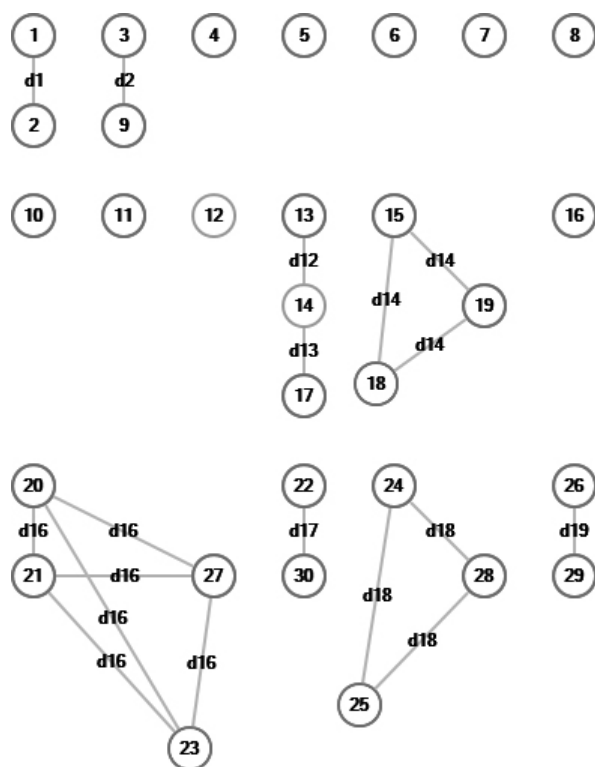


Рис. 6. Граф тестовых заданий теста промежуточного контроля по дисциплине «Программирование» (разбиение вопросов по темам)

Предложен способ визуализации связей между тестовыми заданиями. Описана программная среда, автоматизирующая процесс визуализации связей между разделами учебной дисциплины и тестовыми заданиями.

Возможности предлагаемых инструментов построения и визуализации графа структуры содержания учебной дисциплины и графа тестовых заданий продемонстрированы на примере анализа конкретного теста промежуточного (зачетного) контроля по дисциплине «Программирование».

Разработанная технология анализа тестов может быть применена для многих учебных дисциплин. Ее программная реализация может войти в качестве компонента в инструментальные системы автоматизации измерений в образовании.

Библиографический список

1. Аванесов, В. С. Нетрадиционные тесты [Электронный ресурс] / В. С. Аванесов // Управление школой – 1999. – № 34. – URL : <http://testolog.narod.ru/Theory4.html> (дата обращения: 03.09.2011)
2. Аванесов, В. С. Принципы разработки содержания теста [Электронный ресурс] / В. С. Аванесов // Управление школой – 1999. – № 38. – URL : <http://testolog.narod.ru/Theory6.html> (дата обращения: 25.05.2011)
3. Аванесов, В. С. Содержание теста [Электронный ресурс] / В. С. Аванесов // Управление школой – 1999. – № 36. – URL : <http://testolog.narod.ru/Theory5.html> (дата обращения: 25.05.2011)
4. Аванесов, В. С. Содержание тестов и тестовых заданий [Электронный ресурс] / В. С. Аванесов // Педагогические Измерения, 2007. – № 3. – URL : <http://testolog.narod.ru/Theory63.html> (дата обращения: 03.10.2011)
5. Асанов, М. О. Дискретная математика: графы, матроиды, алгоритмы [Текст] / М. О. Асанов, В. А. Бараскинский, В. В. Расин. – СПб. : Издательство «Лань», 2010. – 368 с.
6. Белоусов, А. И., Ткачев, С. Б. Дискретная математика [Текст] / А. И. Белоусов, С. Б. Ткачев. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2004. – 744 с.
7. Звонников, В. И., Чельшкова, М. Б. Контроль качества обучения при аттестации : компетентностный подход [Текст] / В. И. Звонников, М. Б. Чельшкова. – М. : Университетская книга ; Логос, 2010. – 272 с.
8. Звонников, В. И., Чельшкова, М. Б. Современные средства оценивания результатов обучения [Текст] / В. И. Звонников, М. Б. Чельшкова. – М. : Издательский центр «Академия», 2009. – 224 с.
9. Ойстин, О. Графы и их применение [Текст] / О. Ойстин. – М. : КомКнига, 2006. – 168 с.

10. Подбельский, В. В., Максименкова, О. В. Разработка тестов по программированию для тестирования в компьютерной форме [Текст] / В. В. Подбельский, О. В. Максименкова // Информатизация образования : материалы Международной научно-практической конференции. – Елец : ЕГУ им. И. А. Бунина, 2011.
11. Хаггард, Г. Дискретная математика для программистов [Текст] / Г. Хаггард, Дж. Шлифф, С. Уайтсайдс. – М. : БИНОМ. Лаборатория знаний, 2010. – 627 с.
12. Maksimenkova O. V., Podbelskiy V. V. Educational tests in “Programming” academic subject development // SYRCoSE 2011. Proceeding of the 5-th Spring / Summer Young Reseachers' Colloquium on Software Engineering. Yekaterinburg, 2011, pp. 88 – 93
13. NET FrameWork overview. – URL : <http://www.microsoft.com/net/>
14. Podbelskiy V.V., Maksimenkova O.V. Programming as a part of the Software Engineering education // Proceedings of the 4-th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2010), 2010, pp. 165–168
15. Windows presentation foundation. – URL : <http://msdn.microsoft.com/ru-ru/library/ms754130.aspx>