

тодическом подходе эти занятия могут проводиться на уровне, максимально приближенном к аудиторным занятиям, а возможно даже на более высоком уровне. Выбор платформы зависит от конкретных требований учебного процесса, а также может быть обусловлен характеристиками телекоммуникационных каналов и используемыми клиентскими платформами средств вычислительной техники.

СПИСОК ЛИТЕРАТУРЫ

1. Кузнецов И. Р., Тимофеев А. В. Электронное обучение студентов электротехнического университета // Изв. МАН ВШ. 2012. № 2 (60). С. 50–53.
2. Башарин С. А., Федоров В. В. Теоретические основы электротехники. 4-е изд. М.: Академия, 2010.
3. Разработка и апробация модуля по ДПО «Технология и диагностика тонкопленочных солнечных модулей на основе кремния» в рамках модели смешанного обучения (e-learning) / В. П. Афанасьев, Г. А. Коноплев, Е. И. Теруков, А. В. Тимофеев // Изв. МАН ВШ. 2012. № 2 (60). С. 90–97.

S. A. Basharin, A. V. Timofeev

INTERACTIVE E-LEARNING TECHNOLOGIES IN UNIVERSITIES

In the article some interactive e-learning technologies are offered in institution of higher learning. The choice of different platforms on the base of that the process of the controlled from distance education can be organized is offered.

E-Learning, distance education, virtual laboratories, webinars

УДК 004.434

О. В. Перченко

МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА СИСТЕМЫ ЭЛЕКТРОННОГО ОБУЧЕНИЯ, ОБЕСПЕЧИВАЮЩЕЙ АВТОМАТИЧЕСКУЮ ПРОВЕРКУ ЗАДАЧ

Рассмотрена модель жизненного цикла системы электронного обучения, обеспечивающей автоматическую верификацию решений задач, а также система электронного обучения WiseTasksGeometry, реализующая данную модель для предметной области геометрических задач на построение. Приведено описание предметно-ориентированного языка записи условий, дан пример задачи.

Система электронного обучения, предметно-ориентированный язык, система динамической геометрии, модель жизненного цикла программного обеспечения

Существующие системы динамической геометрии – «Живая математика» (The Geometer's Sketchpad), Cabri, Cinderella, GeoGebra [6], 1С Математический Конструктор (1С МК) и др. основаны на выполнении ряда геометрических построений, преобразований и измерений. Все они, кроме последней (1С МК), не имеют инструментов для проверки решений (в Cabri есть возможность проверять выполнение простейших отношений, например, параллельность линейных объектов). В 1С МК автор задачи может построить предварительно ее решение (задать линейный алгоритм построением динамического чертежа). После этого ученик может проверить свое решение, сравнив его динамически с эталонным (т. е. решение будет признано правильным даже при ином алгоритме решения) [1].

В то же время, за рамками рассмотрения остается проблема адекватного описания задачи. Получается, что описанием задачи служит текст, никак не привязанный к задаче инструментально. Поэтому достаточно сделать ошибку в формулировке, как правильное решение будет оценено как ошибочное, поскольку фактически сравнивалось с решением другой задачи.

В статье предложен иной подход к разработке системы поддержки составления и решения задач, когда описание условий задачи строится на специально разработанном предметно-ориентированном языке, так что алгоритм решения, подготовленный студентом, может быть верифицирован на формальном описании задачи. Задачи, обладающие таким свойством, называются самопроверяемыми [1]. Приведен пример реализации такой системы для предметной области геометрических задач на построение.

Предметно-ориентированный подход к составлению задач по комбинаторике был описан в [2]. Отличием представляемой системы является иной язык описания условий задач (в работе [2] использовалось специально разработанное xml-описание задач) и иной способ верификации решений.

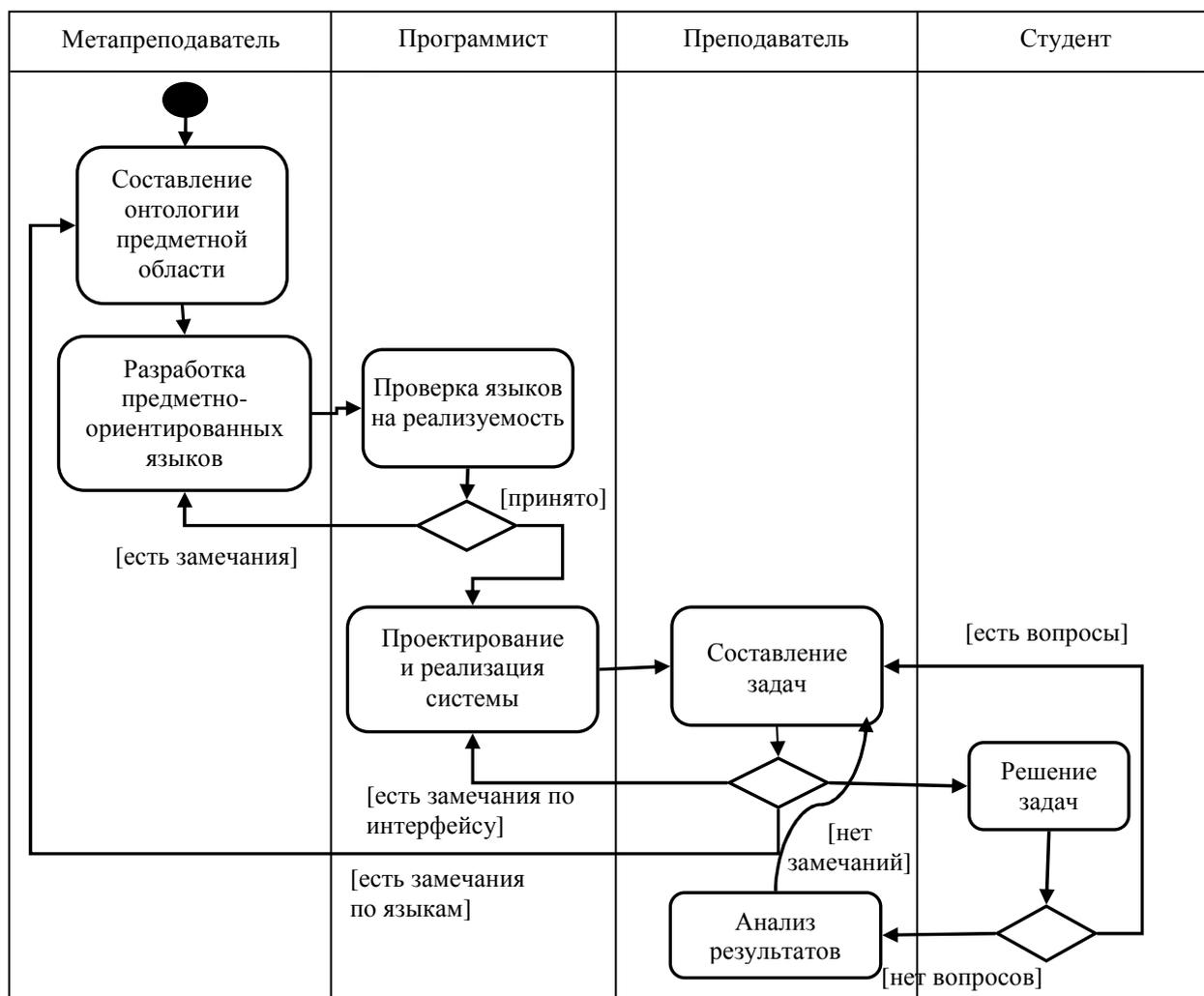


Рис. 1

Описание действий модели приведено в таблице.

1. Модель жизненного цикла системы электронного обучения, поддерживающей самопроверяемые задачи, изображена на рис. 1 в форме UML-диаграммы деятельности [3]. В данной диаграмме действуют следующие актеры:

- метапреподаватель – опытный преподаватель, который глубоко владеет предметной областью и является инженером по знаниям (умеет строить онтологии);

- программист – разработчик системы электронного обучения, который взаимодействует в процессе работы с метапреподавателем и преподавателем;
- преподаватель – пользователь системы электронного обучения, который составляет задачи для системы, анализирует результаты их решения, и имеет обратную связь с программистом и метапреподавателем;
- студент – пользователь системы электронного обучения, который решает в ней задачи и имеет обратную связь с преподавателем.

Действие	Описание
Составление онтологии предметной области	Метапреподаватель составляет на основе своих знаний о предметной области ее онтологию, которая будет лежать в основе предметно-ориентированных языков описания условия и решения. При этом он принимает во внимание замечания преподавателя, возникшие в результате эксплуатации системы
Разработка предметно-ориентированных языков	Метапреподаватель разрабатывает предметно-ориентированные языки описания условий задач и описания решений задач на основе онтологии предметной области (см. п. 2), онтологии абстрактных задач, и абстрактной модели языка [1]
Проверка языков на реализуемость	Программист проверяет возможность реализации языков описания условий задач и описания решений задач. Если у него есть замечания, он сообщает их метапреподавателю, иначе приступает к проектированию и реализации новой системы или модификации существующей
Проектирование и реализация системы	Программист проектирует, реализует и тестирует новую систему электронного обучения или модифицирует существующую систему. При этом он принимает во внимание замечания преподавателя, возникшие в результате эксплуатации системы
Составление задач	Преподаватель составляет новые задачи на предметно-ориентированном языке описания условий (или корректирует ранее составленные задачи), принимая во внимание вопросы студентов, возникшие у них при решении ранее составленных задач, и выводы, сделанные им в процессе анализа результатов решения задач студентами. При наличии замечаний по языку преподаватель сообщает их метапреподавателю. При наличии замечаний по интерфейсу преподаватель сообщает их программисту. При отсутствии замечаний преподаватель дает студентам задание, состоящее в решении задач с помощью системы
Решение задач	Студент решает в системе электронного обучения задачи, составленные преподавателем. Если у него возникают вопросы в процессе решения, он задает их преподавателю, иначе вызывает автоматическую верификацию решений, результаты которой становятся доступными преподавателю
Анализ результатов	Преподаватель анализирует результаты автоматической верификации решений студентов. Выводы, сделанные в ходе анализа, используются преподавателем для составления новых задач или корректировки ранее составленных задач

Представленная модель жизненного цикла (рис. 1) позволяет спроектировать систему электронного обучения на основании онтологии предметной области (см. п. 2), онтологии абстрактных задач и модели абстрактного предметно-ориентированного языка для описания условий задач [1].

2. Онтология и предметно-ориентированный язык геометрических задач. Согласно описанной в п. 1 модели жизненного цикла была разработана система электронного обучения WiseTasksGeometry (wise tasks – умные задачи) [4], обеспечивающая составление, решение и автоматическую проверку задач по геометрии.

В процессе разработки системы была составлена онтология предметной области для задач по геометрии на построение. Онтология создана в системе Protégé. Иерархия классов онтологии приведена в сокращенном виде:

- Thing
 - AbstractGeoElement
 - GeoAngle
 - GeoCircle
 - GeoLine
 - GeoRay
 - GeoSegment
 - GeoPoint
 - GeoPredicate
 - GeoAngleBinPred
 - GeoLineBinPred
 - GeoLineGeoCircleBinPred
 - GeoPointPred
 - GeoSegmentBinPred
 - ValuePred
 - GeoStatement
 - GeoTool
 - AngleFixedTool
 - AngleThreePointsTool
 - CircleCenterPointTool
 - CircleCenterRadTool
 - LineParallTool
 - LinePerpendTool
 - LineTwoPointsTool
 - MidPointTool
 - PointTool
 - RayTwoPointsTool
 - SegmentFixedTool
 - SegmentTwoPointsTool

Классы верхнего уровня данной онтологии представляют собой абстрактную ссылку на геометрический элемент `AbstractGeoElement`, геометрический элемент `GeoElement`, предикат `GeoPredicate` и инструмент `GeoTool`. На следующих уровнях происходит конкретизация.

В соответствии с абстрактной грамматикой предметно-ориентированного языка описания условий [1] был разработан предметно-ориентированный язык записи геометрической задачи. Полученная грамматика в расширенной форме Бэкуса-Наура приведена ниже со следующими соглашениями:

`::=` – отделяет название лексемы от ее описания;

`{A}` – ноль или более элементов A;

[A] – элемент A может входить или не входить;

(A B) – группа элементов;

A|B – либо A, либо B.

Сокращенная грамматика:

Описание ::= Заголовок ТекстовоеУсловие Инструменты Предикаты;

Заголовок ::= “Заголовок: ” Текст;

ТекстовоеУсловие ::= “Условие: “ Текст;

Инструменты ::= ”Инструменты:” {Инструмент} | “Не ограничивать ученика”;

Предикаты ::= {Предикат};

Инструмент ::= “Прямая по двум точкам” | ”Параллельная прямая” | ”Точка” | ”Перпендикулярная прямая” | ”Окружность по центру и радиусу” | ”Середина отрезка” | ”Отрезок по двум точкам” | ”Угол заданной величины” | ”Окружность по центру и точке” | ”Угол по трем точкам” | “Луч по двум точкам”;

Предикат ::= ТочкаЛежитНаПрямой | ПрямыеИлиОтрезкиПараллельны | ПрямыеИлиОтрезкиПерпендикулярны | СерединаОтрезка | ОтрезкиРавны | УглыРавны | ТочкаЛежитНаОтрезке | ТочкаЛежитНаОкружности | ПрямаяКасаетсяОкружности | ВеличинаУгла | ДлинаОтрезка.

3. Программная реализация

3.1. Структура системы

Система WiseTasksGeometry включает модули учителя и ученика, реализованные принципиально в разных концепциях. В модуле учителя автор задачи описывает условие в виде текста, предикатов и логических связей между ними в привычной для себя форме, строит исходный чертеж и задает набор инструментов, которым сможет пользоваться ученик при решении задачи. В модуле ученика студент может прочитать текстовое условие, выполнить необходимые построения и отослать решение на контроль, который состоит в проверке заданных учителем предикатов.

Поддержка динамической геометрии необходима для построения чертежей к задачам. Реализация собственной системы динамической геометрии трудоемка, но существует немало уже реализованных систем. Необходимо было исследовать существующие системы и выбрать из них ту, которая реализована на платформе Java, имеет открытый исходный код, все необходимые возможности и хороший пользовательский интерфейс. Из свободных систем динамической геометрии, реализованных на Java, в конце концов мы остановились на двух: GeoGebra и CaRMetal. Предпочтение было отдано первому варианту, потому что CaRMetal, во-первых, не устроил нас своим интерфейсом: он имеет слишком много элементов управления и разнообразных панелей с настройками, которые, хоть и позволяют сделать массу невозможного для GeoGebra, но при этом пугают пользователей. Во-вторых, при попытке выделить редактор из CaRMetal, чтобы воспользоваться им в разрабатываемой системе, оказалось, что это выделение связано с определенными сложностями, которых не обнаружилось при выделении редактора из GeoGebra. Программа GeoGebra написана Маркусом Хохенвартером на языке Java (работает на многих операционных системах), переведена на 39 языков [6].

3.2. Пример задачи

Для иллюстрации структуры системы приведен пример задачи: «Постройте квадрат ABCD со стороной, равной отрезку KL, заданному на чертеже. AC и BD – диагонали квадрата».

Задача в среде учителя и ученика приведена соответственно на рис. 2 и 3.

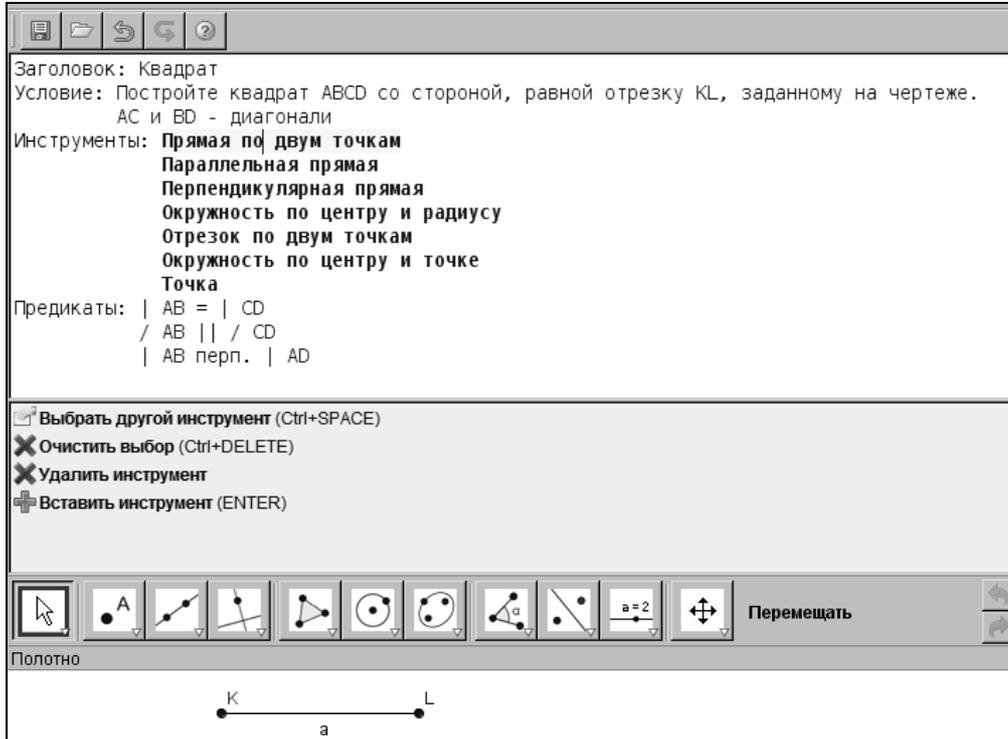


Рис. 2

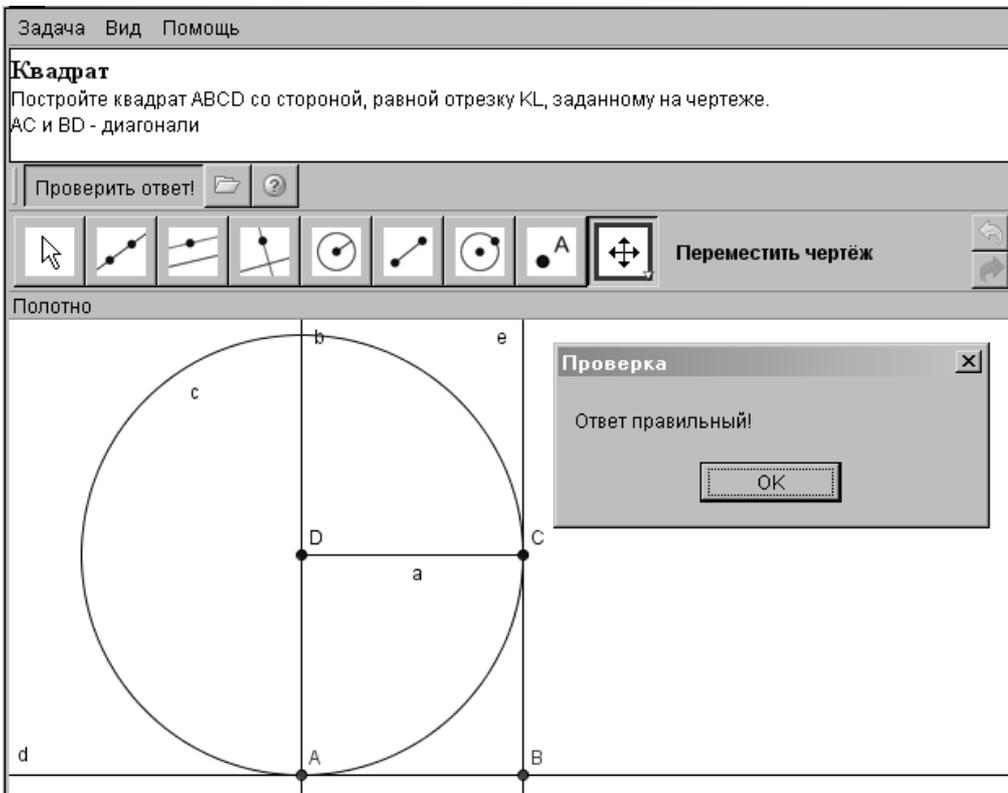


Рис. 3

Модуль учителя состоит из предметно-ориентированного редактора (см. рис. 2, верх), в котором автор вводит условие, и области построения исходного чертежа (см. рис. 2, низ). Редактор создан по образу и подобию системы метапрограммирования MPS [5].

Модуль ученика состоит из текста задачи (см. рис. 3, верх) и области построения решения задачи (см. рис. 3, низ). На рис. 3 показана правильно решенная учеником задача, причем концы отрезка KL с исходного чертежа переименованы в точки D и C.

3.3. Поддерживаемые примитивы

В системе разработаны инструменты для включения в интерфейс ученика построения точек, отрезков, углов, параллельных и перпендикулярных прямых, окружности, середины отрезка. Список инструментов приведен ниже.

- Прямая по двум точкам
- Параллельная прямая
- Точка
- Перпендикулярная прямая
- Окружность по центру и радиусу
- Середина отрезка
- Отрезок по двум точкам
- Угол заданной величины
- Окружность по центру и точке
- Угол по трем точкам

В формальном условии задачи можно описывать отношения между базовыми геометрическими объектами, такими как точка, окружность, отрезок, луч, прямая. Возможны ссылки как на объекты исходного чертежа, так и на объекты, которые необходимо построить ученику.

В системе реализованы следующие отношения между объектами: принадлежность точки прямой, окружности, параллельность и перпендикулярность линейных объектов, принадлежность точки середине отрезка, равенство отрезков и углов, касание прямой и окружности. Приведем список предикатов.

- Точка лежит на прямой
- Прямые или отрезки параллельны
- Прямые или отрезки перпендикулярны
- Середина отрезка
- Отрезки равны
- Углы равны
- Точка лежит на отрезке
- Точка лежит на окружности
- Прямая касается окружности
- Величина угла
- Длина отрезка

В статье рассмотрена модель жизненного цикла системы электронного обучения, обеспечивающей автоматическую верификацию решений задач, а также система электронного обучения WiseTasksGeometry, реализующая данную модель для предметной области геометрических задач на построение.

Система WiseTasksGeometry внедрена в учебный процесс средней школы № 521 Красногвардейского района г. Санкт-Петербурга в качестве инструмента контроля знаний по теме «Построение с помощью циркуля и линейки» в курсе геометрии восьмого класса, а также на курсах повышения квалификации учителей в учебном центре ООО «ИНТОКС».

СПИСОК ЛИТЕРАТУРЫ

1. Генерация математических задач и верификация решений в автоматизированных системах поддержки обучения / Д. И. Манцеров, О. В. Перченко, С. Н. Поздняков и др. СПб.: Изд-во СПбГЭТУ ЛЭТИ, 2012.
2. Богданов М. С. Автоматизация проверки решения задач по формальному описанию ее условия // Компьютерные инструменты в образовании. 2006. № 4. С. 51–57.
3. Леоненков А. В. Самоучитель UML 2. СПб.: БХВ-Петербург, 2007.
4. Поздняков С. Н., Перченко О. В., Посов И. А. Автоматизация проверки решения геометрических задач по описанию их условий на предметно-ориентированном языке // Компьютерные инструменты в образовании. 2012. № 1. С. 37–44.
5. Сайт программы MPS: <http://www.jetbrains.com/mps/> (дата обращения: 12.04.2013)
6. Сайт системы динамической геометрии GeoGebra: <http://www.geogebra.org> (дата обращения: 12.04.2013)

O. V. Perchenok

LIFECYCLE MODEL OF E-LEARNING SYSTEM WHICH PROVIDES AUTOMATIC PROBLEM VERIFICATION

The article describes lifecycle model of e-learning system which provides automatic problem verification. WiseTasksGeometry e-learning system, which implements the model for the geometrical constructional problems domain, is also depicted. The domain-specific language of geometrical problem is outlined, the problem example is given.

E-learning system, domain-specific language, dynamic geometry system, software lifecycle model