9 НОВЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 004.434

ПРЕДМЕТНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ В ОПИСАНИИ ЗАДАЧ НАУЧНЫХ СОРЕВНОВАНИЙ

М.С. Богданов, С.Е. Рукшин, Д.О. Степуленок

Разработана технология разработки предметно-ориентированных языков для представления задач математического характера в системах дистанционного обучения, которые для своей проверки не требуют ввода ответа методистом, т.е. обладают свойством «самопроверяемости».

Ключевые слова: предметно-ориентированные языки, представление математических задач, системы дистанционного обучения.

Введение

Развитие дистанционного обучения (ДО) сделало актуальным автоматизацию поддержки ученика при решении им задач математического характера, что актуально для тех предметов, которые используют язык математики. В то же время системы дистанционного обучения оснащены этими возможностями в минимальной степени, а именно, они позволяют проверять ответы для небольшого числа форматов, таких как выбор ответа из нескольких предложенных, строковый формат, числовой формат и некоторых других, и принимать решения без их автоматической проверки в виде текста (гипертекста). Тем самым существующие дистанционные системы обучения предполагают достаточно большую работу методиста при подготовке задач:

- переформулировка задач в приемлемую для системы форму;
- решение задач;
- занесение ответов в систему;
- ручную проверку решений задач, решения которых представлены в «свободной» форме.

Кроме достаточно большой работы методиста на разных этапах подготовки учебных материалов, следует обратить внимание на дополнительные обстоятельства, такие как:

- необходимость проверки введенных ответов (поскольку сравнение с ними будет происходить автоматически);
- защиту введенных ответов от «взлома» с целью введения правильного ответа без попыток выполнить решение задачи (в существующих системах ДО проверяются не решения задачи, а только ответы).

Таким образом, актуальной является задача поиска таких средств для описания задач в системах ДО, которые были бы свободны от всех, либо, по крайней мере, от части отмеченных недостатков.

Авторы предлагают следующий путь решения этой задачи:

- разработать средства для формального описания задач, которые бы позволили вообще отказаться от ввода ответов методистами: задачи будут обладать свойством «самопроверяемости»;
- разработать интерфейсы для основных типов математических задач, которые позволят методистам использовать для описания задач предметно-ориентированные языки, имитирующие привычные языки и средства предметной области;
- разработать средства, обеспечивающие конструирование только математически корректных задач; это позволит использовать разработанные средства не только для создания учебных материалов для поддержки школьного курса или занятий кружкового характера и олимпиад, но и для постановки и исследования нерешенных задач.

В работах [1–3] показана эффективность такого подхода для работы с задачами по комбинаторике. В этой работе предложенные идеи обобщаются и иллюстрируются на примере создания системы для работы с геометрическими задачами на «геометрическом языке».

Среда WiseTasks для поддержки «самопроверяемых» задач по комбинаторике

Для того чтобы сформулировать задачу в общем виде, рассмотрим особенности ее решения на одном примере: работе с комбинаторными задачами.

В описании комбинаторной задачи в системе выделяются две ее основные части. Первая предназначена для представления задачи в понятной человеку форме (например, словесное описание условия задачи), вторая представляет формализованное описание, понятное машине, вычислительной среде или интерпретатору, которые собственно и осуществляют проверку решения.

Формализованное описание в системе комбинаторной задачи в общем случае состоит из описания некоторого множества и определенного на нем предиката. Условием задачи может быть либо требование определить все элементы множества, удовлетворяющие данному предикату, либо привести первый удовлетворяющий элемент в соответствии с определенной нумерацией, либо, что встречается чаще всего, посчитать количество удовлетворяющих предикат элементов. Например, известная задача о количестве счастливых билетов может быть сформулирована как задача на множестве наборов из шести цифр, где предикатом является утверждение, что сумма первых трех цифр равна сумме последних трех цифр, а найти необходимо количество удовлетворяющих предикат элементов.

Рассмотрим описание задачи в системе на примере задачи о счастливых билетах. Внутренним представлением задачи для системы является XML-дерево, структурирующее данные об условии. Рассмотрим содержимое элемента mathDescription, содержащего формальное описание условия. Он состоит из двух подэлементов, sourceSet и verifier, содержащих, соответственно, описание множества и метода проверки ответа участника. Элемент sourceSet содержит в себе элемент set, имеющий атрибут type – тип множества, который может быть множеством перестановок, размещений, комбинаций, отрезком числовой прямой, а так же декартовым произведением, в последнем случае элемент set содержит в себе дочерние set элементы.

Элемент verifier описывает верификатор, он имеет атрибут type, определяющий один из описанных выше типов верификатора. Для задачи о счастливых билетах используется тип SimpleVerifier, соответствующий подсчету количества элементов множества, удовлетворяющих предикату. Сам предикат описывается в элементе function. Этот элемент соответствует композиции функций и может содержать дочерние function элементы. Конкретная используемая функция определяется по атрибуту type. Система предоставляет большой выбор функций, как простейших (логических и арифметических), так и более сложных.

На рис. 1 представлено формальное описание задачи, позволяющее осуществить самопроверку решения этой задачи системой. XML-элементы представлены в виде дерева и в читаемом варианте, в частности, элементы function записываются заглавной буквой F вместе с приписанным атрибутом типа. Отметим, что в системе учтена возможность параметризации условия, когда по одному такому описанию среда генерирует разные «клоны» исходной задачи. Эта часть мало зависит от способа формализации решения для компьютера и может состоять из перечисления параметров и их значений. Параметризация видна на приведенном примере в элементе descriptions-params.

Система допускает несколько режимов создания самопроверяемых задач:

1. описание задачи в формате XML, используя разработанный синтаксис языка (на практике не используется);
2. использование редактора общего назначения (изображен на рис. 1), помогающего соблюдать разработанный синтаксис (используется автором языка);
3. использование предметных сред-интерфейсов, в которых строится описание задач основных типов и которые затем автоматически конвертируются в разработанный формат XML (используется преподавателями и учениками).

Предметно-ориентированные языки программирования как средство описания задач

Перейдем теперь к решению поставленной задачи в общем виде. Необходимо уменьшить разрыв между недоступным учителям универсальным редактором и редакторами узких классов задач. Для этого потребуется привлечь концепцию предметно-ориентированных языков программирования.

Предметно-ориентированные языки являются основой парадигмы языкоориентированного программирования. Его сутью является то, что при разработке программного обеспечения та часть, которая требует большего знания предметной области, чем особенности платформы, реализуется на специально созданном языке или языках. Язык исполняется в рамках предметно-ориентированной платформы. Предметно-ориентированный язык разрабатывается так, чтобы быть максимально доступным для специалистов предметной области, часто не являющихся программистами. В данной работе в качестве специалистов предметной области рассматриваются учителя или методисты, составляющие задачи для использования в олимпиадах или организации иной деятельности учащихся, связанной с решением задач.

В настоящее время наиболее развитой и перспективной средой для создания предметно-ориентированных языков программирования является система MPS, концепции которой и стали основой для создания нужного инструментария [4].

Система MPS позволяет создать предметно-ориентированный язык вместе со средством его редактирования. Создаваемый язык может быть расширением уже существующего, например, быть расширением языка Java, или же создаваться полностью с нуля, что более интересно для нас, потому что предполагаемыми пользователями языка являются преподаватели и методисты, незнакомые с Java. При создании языка описываются возможные узлы его абстрактного синтаксического дерева и способ их отображения на экране. При вводе текста на созданном языке вводится сразу же его дерево разбора, что позво-

ляет избежать стадии синтаксического анализа и навязывает пользователю правильный синтаксис, не позволяя вводить неверные конструкции. Редактор языка напоминает обычный текстовый редактор, но на каждой стадии ввода подсказывает, какую именно информацию и как необходимо ввести пользователю, при этом предоставляет возможные варианты дополнения текста. Хорошо разработанный предметно-ориентированный язык может использоваться без предварительного обучения потому, что уже после запуска редактора пользователь видит, какую именно информацию от него требуется ввести. Создаваемые редакторы имеют больше возможностей, чем оконные интерфейсы с визуальными элементами управления, потому что более динамичны и требуют меньшего пространства для большего количества информации. К сожалению, на данный момент система MPS не позволяет использовать созданные языки вне своего интерфейса и, хотя и распространяется бесплатно, является слишком громоздкой, чтобы предлагать учителям ее использовать. По этой причине при создании предметно-ориентированных языков для учителей приходится пользоваться собственными средствами.

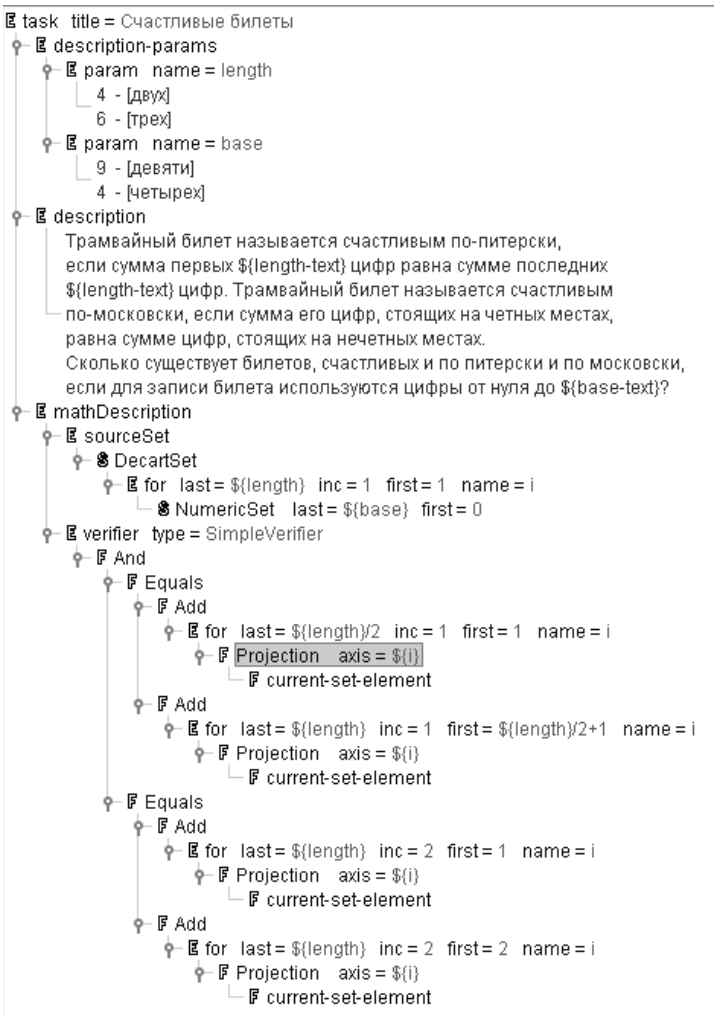

Рис. 1. Описание задачи о счастливых билетах в форме XML дерева

Сформулируем основные этапы разработки предметно-ориентированных языков для систем педагогического назначения.

1. Ограничение предметной области, т.е. сужение множества задач определенного раздела математики до рамок, в которых возможна автоматическая проверка решений по условию.
2. Выделение понятий предметной области, необходимых для описания задач и возможных отношений между понятиями.
3. Попытка описать задачи предметной области на языке понятий и связей между ними, добавление недостающих понятий и определение синтаксиса.
4. Создание объектной модели понятий и связей между ними.
5. Создание редактора предметно-ориентированного языка, т.е. описание редакторов для объектов модели, которые позволят переводить вводимые условия задач в данные объектной модели.
6. Реализация модуля проверки решений.

Создатель предметно-ориентированного языка для описания редакторов объектов модели может использовать понятие клеток, существующее в MPS. Редактор для объекта – это некоторое количество клеток, которые отображают информацию о редактируемом объекте. Например, объект «отрезок натуральных чисел» может иметь два поля, начальное и конечное число. Для настройки отображения объекта с помощью клеток необходимо добавить объекту описание, что его редактор состоит из четырех клеток: на первой написано слово «от», на второй, расположенной справа, написано значение соответствующего поля, далее, в клетке написано «до» и далее – значение другого поля объекта. Клетки могут быть вложены друг в друга, например, объект может содержать в себе подобъект, который редактируется с помощью клетки, состоящей из клеток подобъекта, тем самым клетки образуют иерархическую структуру.

В качестве примера рассмотрим применение предложенной технологии для создания среды, поддерживающей формулировку геометрических задач учителем, решение задач учеником, верификацию решений системой.

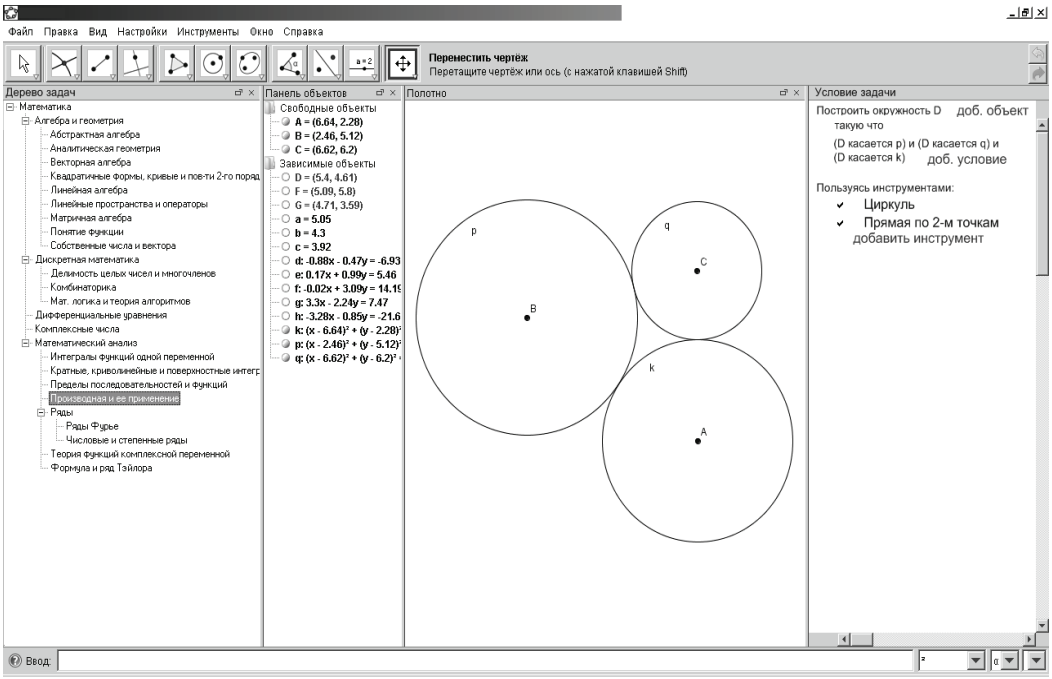

Рис. 2. Общий вид «учительского» интерфейса для геометрических задач

На рис. 2 изображен интерфейс учителя по созданию задачи. Левая панель содержит дерево задач, загруженных в систему. Средние две панели являются геометрическим редактором, правая содержит чертеж, а левая – список объектов чертежа. Самая правая панель – это редактор текста на предметно-ориентированном языке.

Для составления задачи на построение преподаватель рисует исходное построение для условия, которое будет показано ученику перед решением. Кроме того, он пишет в специальном редакторе условие задачи, пользуясь геометрическими обозначениями, например, символами параллельности и перпендикулярности. На рис. 2 приведен пример задачи: дано три окружности, касающиеся друг друга, постройте четвертую, касающуюся трех данных.

Для описания данной задачи в языке существует элемент *окружность*, а также предикат *касаться*. Тогда при начальном построении мы нарисуем три касающиеся друг друга окружности, а в качестве условия задачи укажем, что нужно построить еще одну окружность, которая будет касаться указанных трех. Допустим, исходные окружности обозначены буквами *A*, *B* и *C* (обозначения мы задаем в визуальном геометрическом редакторе), тогда условие выглядит так: Построить окружность *D*, такую что *(D касается A) и (D касается B) и (D касается C)*. Эти предикаты, а также набор инструментов для построения, требуемый объект для построения и текст условия задачи, задаются в редакторе предметно-ориентированного языка. Итак, геометрические задачи на построение, описанные с помощью предложенного геометрического предметно-ориентированного языка, имеют схожую форму условия: Построить [перечисление объектов для построения] так, чтобы [Предикат] используя инструменты [Список инструментов]. Учителю необходимо лишь заполнить пропуски, при этом на каждом этапе ввода условия пользователь выбирает пропуск для заполнения и один из возможных вариантов заполнения. Расположение текстов и пропусков на экране задается разработчиком интерфейса с помощью описанного выше механизма клеток.

Интерфейс ученика отличается от интерфейса учителя тем, что вместо формального текста условия он видит условие на естественном языке и ему доступна кнопка «отправить решение на проверку».

Заключение

Представленный в статье подход позволяет по-новому подойти к проблеме организации научных соревнований. Разработка адекватных задаче средств ее представления в системе, языков описания таких задач, позволяет исключить этап предварительного решения задач перед размещением ее в системе дистанционного обучения. Задача, описанная на таком языке, обладает свойством «самопроверяемости», что позволяет перейти от задач с «известным ответом» к использованию исследовательских задач.

Например, расширение сред динамической геометрии с помощью таких языков, позволяют не только поддержать конструктивную деятельность ученика в среде, но позволить учителю или самому ученику ставить новые задачи, используя язык, аналогичный тому, который используется при описании решений задач.

Литература

1. Богданов М.С. Автоматизация проверки решения задачи по формальному описанию ее условия // Компьютерные инструменты в образовании. – 2006. – № 4. – С. 51–57.
2. Pozdnyakov S., Bogdanov M., Pukhov A. Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics // The 9th International conference «Teaching Mathematics: Retrospective and Perspectives». Vilnius Pedagogical University. – 2008, 16–17 May.
3. Богданов М.С., Дубров С.Ю., Поздняков С.Н. Декларативная модель формального описания математических задач в системе проведения олимпиад WISETASKS и ее автоматическая проверка // Труды научно-технической конференции «Компьютерное моделирование 2009», 23–24 июня 2009. – СПб: Изд-во Политехнического университета, 2009. – С. 179–180.
4. Dmitriev Sergey. Language Oriented Programming: The Next Programming Paradigm. [Электронный ресурс]. – Режим доступа: http://www.jetbrains.com/mps/docs/Language_Oriented_Programming.pdf, свободный. – Загл. с экрана. – Яз. рус. (дата обращения 18.02.2010).

Богданов Михаил Сергеевич – AXIOM SL Inc., программист, flown@mail.ru
Рукшин Сергей Евгеньевич – Российский государственный педагогический университет им. А.И. Герцена, кандидат физ.-мат. наук доцент, vliuser@gmail.com
Степуленок Денис Олегович – Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», ассистент, super.denis@gmail.com