



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004. 832.2

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЧЕСКОГО РЕШЕНИЯ ЗАДАЧ ПО ПЛАНИМЕТРИИ

Курбатов С.С., Лобзин А.П., Хахалин Г.К.

Научно-исследовательский центр электронной вычислительной техники, Москва, Россия

curbatow.serg@yandex.ru

lobzin@rambler.ru

gkhakhalin@yandex.ru

В статье рассмотрена система автоматического решения задач по планиметрии с использованием онтологии и естественно-языкового интерфейса. В рамках системы разработано программное обеспечение, использующее онтологию при поиске решения задачи, сформулированной на предметно-ориентированном естественном языке. Для найденного решения разработана визуализация на предметном и онтологическом уровнях. Программное обеспечение включает макросы Word для работы с графикой и решатель, реализованный в программной среде онтологии.

Ключевые слова: решение геометрических задач; лингвистическая поддержка; прикладная онтология.

Введение

Исторически геометрия явилась провозвестником аксиоматического метода, нашедшего ныне широкое распространение не только в математике, но и в ряде других наук - физике, химии, биологии, лингвистике. Свою роль в осмыслении таких фундаментальных понятий как аксиоматика, формальная система, непротиворечивость, модель, интерпретация, геометрия не утратила и на современном этапе. Это обусловлено большой наглядностью геометрических построений, обеспечивающей возможность опираться на интуицию. Последняя не может заменить доказательство, но существенно облегчает его понимание. В современных условиях, учитывая графические возможности ЭВМ, геометрия приобретает дополнительную значимость. Проектируются графические системы обучения геометрии и решения задач, существенно повышающие качество образования [Давыденко и др., 2011].

Интерес к автоматическому решению задач и доказательству теорем, относящихся к геометрии, возник уже на раннем этапе исследований в области искусственного интеллекта (ИИ). В ранних работах был четко сформулирован отказ от полного перебора в пользу использования эвристических методов [Геллернтер, 1967]. В дальнейшем работы этого плана не получили достаточного развития и внимание было уделено представлению знаний,

экспертным системам, обработке естественного языка и т.п. В данной работе описывается программное обеспечение (ПО) системы автоматического решения задач по планиметрии с использованием онтологии и естественно-языкового интерфейса. Интеграция лингвистических методов, возможностей онтологии для представления знаний предметной области и современной графики позволяет по-новому взглянуть на роль геометрии в исследованиях по искусственному интеллекту и на возможности компьютерного обучения.

Эвристики, используемые при разработке данного ПО, базируются на более высоком концептуальном уровне, чем применяемые в [Геллернтер, 1967]. Такой уровень обеспечивается возможностями онтологии и средствами обращения к синтезируемому чертежу. Элементы высокоуровневых эвристик, конкретизируемых в данной предметной области (планиметрия), намечены в [Курбатов, 2010]. Высокий уровень эвристик позволяет не только существенно сократить перебор, но и предоставляет возможность пояснить процесс получения решения. Средства взаимодействия ПО с онтологией позволяют в диалоге обосновать выбор некоторой эвристики и графически отобразить по шагам путь к найденному решению. Отчасти эти средства реализуют концепцию решения задач, развитую в классической работе [Пойя, 1970], в которой описан процесс выдвижения эмпирических догадок, используемых в дальнейшем для строгого доказательства.

1. Структура системы

Система автоматического решения задач по планиметрии с использованием онтологии и естественно-языкового интерфейса представляет собой специализацию общей системы, в комплексе интегрирующей возможности анализа/синтеза естественного языка и анализа/синтеза изображений на базе онтологии [Khakhalin at al, 2012]. Блок-схема системы приведена на рис.1.

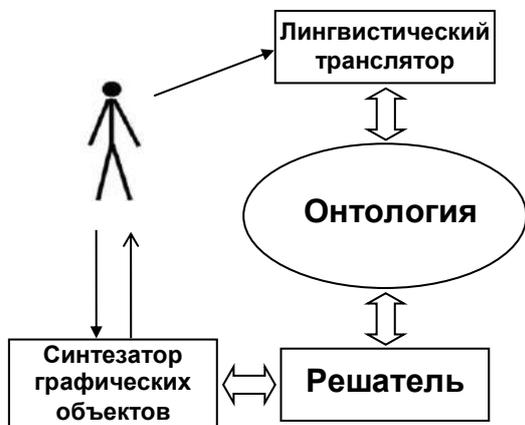


Рисунок 1 – Блок-схема системы

Пользователь формулирует планиметрическую задачу на предметно-ориентированном естественном языке (ЕЯ). Лингвистический транслятор переводит эту формулировку в онтологическую структуру, в которой специальным образом помечены известные элементы (дано) и неизвестные (требуется найти; эти концепты помечаются вопросами).

Задача решателя - с помощью базовых операций означить неизвестные элементы. Для этого он расширяет входную структуру, используя аксиоматику предметной области, эвристики и базовые операции преобразования с целью снятия вопросов. В случае успешной работы решателя формируется финальная расширенная онтологическая структура решения задачи, которая графически отображается пользователю.

Задача синтезатора графических объектов – визуализировать решение, что позволяет пользователю наглядно видеть весь процесс решения задачи и его графический результат с возможностью пошагового отображения.

Задача лингвистического транслятора - по ЕЯ-тексту планиметрической задачи построить концептуальную структуру описания ситуации в терминах понятий и отношений прикладной онтологии, которую решатель использует для поиска ответа с помощью выполнения базовых операций.

Обработка ЕЯ-текстов планиметрических задач основана на лингвистической концепции перифразирования [Апресян и др., 1989], адаптированной для поставленных в работе целей.

С помощью правил перифразирования предметно-ориентированный текст переводится в канонические структуры, непосредственно отображаемые в онтологию.

Работу реализованного алгоритма лингвистического анализа в текущей версии удобно описать на примере конкретной задачи в рассматриваемой области. Более подробно работа лингвистического транслятора описана в [Курбатов и др., 2015]

Формулировка задачи: *Построить окружность, проходящую через две заданные точки, с центром на заданной прямой.*

Опустим этапы морфологического и синтаксического анализа (хотя они и не являются тривиальными). В результате выполнения соответствующих правил перифразирования текст задачи представляется в виде следующего (канонического) описания:

Построить произвольную точку А. Построить произвольную точку В.

Построить произвольную прямую. Построить окружность.

Точка А принадлежит окружности. Точка В принадлежит окружности.

Центр окружности принадлежит прямой.

Текст канонического описания непосредственно отображается в структуру онтологии, где помечены как "известны" (или "дано") *точка А, точка В и прямая. Точка С и окружность* помечены как "неизвестны" (или "найти"). Графическое представление этой онтологической структуры дано на рис. 2.

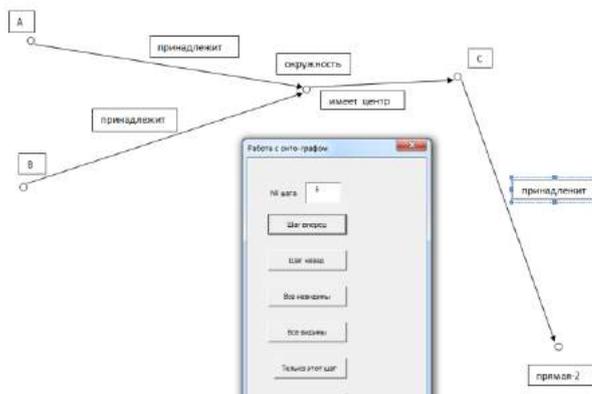


Рисунок 2 – Онтологическая структура задачи

После успешной лингвистической трансляции в онтологической структуре фиксируются объекты и отношения, определяющие условия задачи и требуемый результат. Онтологический решатель должен найти последовательность базовых операций, позволяющих построить требуемый объект (*точку, окружность, прямую* и т.д.). В качестве базовых операций используются "создать точку" (*отрезок, прямую, окружность*), "найти середину отрезка", "провести перпендикуляр из точки на отрезке прямой", "опустить перпендикуляр из точки на прямую" и т.п.

Набор базовых операций не фиксирован и может пополняться. Кроме того, базовые операции могут быть заблокированы при поиске результата. Это позволяет искать решение в условиях ограничений типа "использовать только линейку и окружность с заданным центром (без циркуля)". Базовые функции реализованы с помощью макросов Word (графические функции) и для их вызова онтология формирует формальный текст, включающий имена функций и имена и значения соответствующих параметров. Стиль вызова отличается от используемого в [Литвинович, 2012], поскольку интерпретация текста макроса выполняется только внутри документа. Примеры графических функций приведены в Таблице 1.

Таблица 1 – Примеры графических функций

Crt_tch (x, y, name, rnd)	Построить имитацию точки с координатами x и y и именем name*.
Crt_otr (P1, P2, rnd)	Построить отрезок с концевыми точками P1 и P2.
Crt_line (... , rnd)	Построить прямую **.
Crt_cyrc (P1, L1, rnd)	Построить окружность по центру(P1) и радиусу(L1).
Rel_line (L1, L2)	Определить отношение прямых L1 и L2.
Peres_line (L1, L2, P)	Построить точку с именем P, являющуюся точкой пересечения прямых L1 и L2.
Rel_LC (L, C)	Определить отношение прямой L и окружности C.
Peres_LC (L, C, P1, P2)	Построить точки пересечения прямой и окружности P1 и P2.
Seredina (L, P)	Построить точку P середины отрезка L.
Perpen (L1, P, L2)	Построить перпендикуляр L2 к отрезку L1 в точке P.
Sravn_otr (L1, L2)	Сравнить отрезки L1 и L2 ***.
Perpen_line (P1, L, PL)	Опустить перпендикуляр из точки P1 на прямую L. Фиксировать результат в виде точки или отрезка PL.
Ctr_ugol (P1, P2, P3, U, rnd)	Построить угол U по трем точкам. P2 - вершина угла.
Crt_biss (U, L)	Построить биссектрису L угла U.
Crt_treug (... , rnd)	Построить треугольник.
Crt_paral (... , rnd)	Построить параллелограмм.
Crt_trap (... , rnd)	Построить трапецию.
Vidim/nevidim (...)	Определить видимость/невидимость графического объекта.

*) Параметр rnd определяет случайное задание графического объекта (точка, прямая, треугольник, трапеция и т.п.).

***) По двум точкам, по принадлежащему ей отрезку или по точке и коэффициенту наклона.

****) Отрезки могут быть равны (конгруэнтны) или могут быть в отношении больше/меньше.

2. Онтология

В онтологии задаются знания о естественном языке (синтаксис, семантика языка, правила перефразирования, а также знания об отображении ЕЯ-описаний в формальные конструкции предметной области), знания о предметной области (концепты, связи, аксиоматика, эвристики) и знания о графике.

2.1. Общее описание онтологии

Нижний уровень онтологии представлен базовыми структурами, задающими структурированную семантическую сеть. Системные отношения сети определяют вложенность фрагментов сети, а также связи "род-вид" и "часть-целое". Внесистемные отношения определяются конкретикой предметной области.

Онтология реализована на инструментальной СУБД Progress и использует ее возможности для интерфейса с онтологией.

Интерфейс с онтологией обеспечивает возможности текстового ввода/вывода онтологических структур и их редактирования. Более высокий уровень работы с онтологией позволяет, в частности, использовать гиперграфы и автоматизировать ввод правил перефразирования с помощью ЕЯ-интерфейса. Полное текстовое описание базовых онтологических структур дано в [Лобзин, 2015].

2.2. Концепты, отношения, поиск решения

В текущей версии системы рассматриваются планиметрические задачи на построение. В качестве концептов выступают базовые понятия (точка, прямая, плоскость, окружность) и производные понятия (отрезок, луч, угол, треугольник и т.д.). В качестве отношений используются отношения принадлежность, пересечение, параллельность, перпендикулярность и т.д. Введение декартовых координат позволяет задавать аксиоматику концептов и отношений в алгебраическом виде. Например, точка задается парой чисел (x, y), прямая - двумя точками (x1, y1) - (x2, y2), а аксиоматика точка A (x, y) принадлежит прямой, если и только если

$$x = x_1 = x_2 \text{ (для вертикальной прямой),}$$

$$y = y_1 = y_2 \text{ (для горизонтальной прямой)}$$

и, наконец, $y = k \cdot x + b$, где $k = (y_2 - y_1) / (x_2 - x_1)$ и $b = y_1 - (y_2 - y_1) / (x_2 - x_1) \cdot x_1$.

Для отрезка (x1, y1) - (x2, y2) аксиоматика записывается так: точка A (x, y) принадлежит горизонтальному отрезку, если и только если точка A (x, y) принадлежит прямой (x1, y1) - (x2, y2) и $x_1 \leq x \leq x_2$ при $x_1 < x_2$ или $x_2 \leq x \leq x_1$ при $x_1 > x_2$.

$> x_2$ (для не вертикального отрезка). Аналогично описание для вертикального отрезка. В онтологии аксиоматика зафиксирована декларативно и поддерживается программно. В частности, известно, что прямые могут *совпадать*, *быть параллельны* или *пересекаться*. Соответствующая программная поддержка позволяет определить какое из данных отношений имеет место для конкретных прямых.

Онтология обращается к чертежу для выполнения графических операций типа "создать точку", "провести прямую", "создать окружность", "создать отрезок" и т.п. Дополнительно онтология может запросить по чертежу эмпирические данные (равенство углов или отрезков, перпендикулярность или параллельность и т.д.) и использовать их для сокращения перебора при поиске решения. Стиль такого взаимодействия с чертежом описан в [Курбатов, 2010].

Поиск решения использует эвристики онтологии для сокращения перебора. В полученной после лингвистической трансляции структуре одни объекты отмечены, как <известны>, а другие как <НЕ-известны>. Задача решателя - путем ввода новых объектов/отношений и использования аксиоматики предметной области добиться, чтобы требуемые в задаче объекты получили статус <известны>. В процессе перебора строится дерево, вершины которого соответствуют текущей онтологической структуре (и соответствующему чертежу). Каждой вершине дерева приписан вес, определяющий <перспективность> вершины.

Перебор всегда ведется с вершины, обладающей максимальной перспективностью. Алгоритм вычисления перспективности зависит от эвристических соображений типа <аналогичность>, <наличие объектов со статусом - известны>, <расстояние до корня дерева> и т.п. В текущей реализации перспективность рассчитывается по формуле: $k \cdot (\sum v_i) - r$, где k - коэффициент базовой операции, v_i - объекты со статусом <известны>, а r - расстояние до корня дерева. Коэффициент k отражает полезность использования базовой операции и в первом приближении определяется статистически.

Подчеркнем, что это именно первое приближение, дерево перебора также организовано как онтологическая структура и в принципе может использовать любые знания из базы.

В рассматриваемых задачах на построение для большинства базовых операций результат выполнения непосредственно отображается на чертеже. Это позволяет наглядно представить не только успешно полученный результат, но и неудачные попытки решения. Отметим, что онтологически-ориентированный перебор существенно абстрагирован от концептуальной структуры. Например, поиск на "и/или" графах является более конкретным, информация о том, что вершины в онтологической структуре связаны отношением "и/или", может быть известно только в

онтологии. В принципе возможен случай, что отношения имеет тип "и/или", но какой именно - неизвестно. В задачи решателя онтологии входит выбор действий по определению типа.

2.3. Интерфейс с онтологией

В текущей реализации ПО интерфейс использует достаточно общий метод взаимодействия с онтологией через текстовые файлы. Интерпретация этих файлов выполняется отдельно на уровне графических функций и в программной среде онтологии. При этом онтология выступает в роли системы, принимающей решение, графические функции - в роли выполняющей (эффеторной) системы. В текстовом файле онтология формирует имена и параметры для вызова базовых функций, интерпретация практически сводится к вызову (*call* функции с соответствующими аргументами). Соответствие ЕЯ-наименований базовых функций и объектов визуализации, а также концептуальное описание чертежа содержатся в онтологии. Пример взаимодействия онтологии и графических функций приведен далее.

Целесообразная организация интерфейса сложна не только в технологическом, но и в идейном плане. Интерфейс должен учитывать баланс между такими факторами как эффективность выполнения, удобство восприятия, гибкость настройки на другую программную среду и т.д. Гибкость настройки требует особого внимания, как отмечено выше, в используемой онтологической структуре неизвестные концепты специальным образом отмечены. Квалифицированный подход к гибкости предполагает, что переход от используемой онтологической структуры (и соответствующей аксиоматики) к RDF-триплетам должен быть относительно простым.

Это означает, что будет возможно использовать ресурсы глобальной семантической сети (semantic web), в частности, представлять онтологическую структуру задачи в виде запроса на SPARQL. В этом случае неизвестные концепты будут помечены знаком вопроса "?", а сама структура должна быть описана на языке OWL. Разумеется, и аксиоматика задачи должна быть соответствующим образом описана. Вопрос о возможностях такого представления и эффективности использования логического вывода в OWL является предметом специального исследования.

3. Визуализация

3.1. Графические объекты и базовые операции

Для графических объектов онтологии реализовано более 30 базовых операций типа: "создать произвольную прямую", "построить отрезок", "найти точку пересечения прямых", "построить из точки касательную к окружности" и т.д. (см. Таблицу 1.). Отрезки и окружности имеют

внутренние имена (*line x, oval y*), но для диалога важны ЕЯ-наименования таких объектов. Поэтому в число базовых включен и ряд сервисных функций, в частности, функция ЕЯ-наименования отрезков и окружностей (фиксируется в свойстве AlternativeText соответствующего объекта).

К сервисным функциям относятся также раскраска объектов, их перемещение по экрану (или вращение), изменение свойства Visible = msoFalse (или msoTrue) и т.п. Изменение последнего свойства позволяет сделать невидимым указанный объект, что повышает наглядность при сложном чертеже. Важной сервисной функцией является определение выделенных пользователем объектов на чертеже (Selection), что дает возможность организовать диалог с онтологией относительно этих объектов.

Базовые функции, включающие слово "произвольный" (произвольная прямая, окружность, треугольник и т.д.), используют случайный выбор при генерации свойств соответствующих объектов. Это позволяет разнообразить чертеж для одной и той же задачи, а также (в перспективе) формировать в онтологии предположения, используя эмпирические данные чертежа.

3.2. Онтологическая и предметная визуализация

В процессе онтологически-ориентированного перебора выполняется ряд попыток. Эти попытки в онтологической структуре отображаются путем ввода новых экземпляров концептов (*точек, прямых, отрезков, окружностей* и т.п.) и отношений между ними. Визуализацию этой структуры будем называть онтологической, она представляет интерес в основном для инженера по знаниям (или администратора онтологии). Для пользователя важна визуализация найденного решения в терминах базовых операций.

Первоначально на чертеже для нашей задачи с помощью базовых операций типа *создать точку* и *создать прямую*, построены только "две точки (А и В)" и "прямая-1".

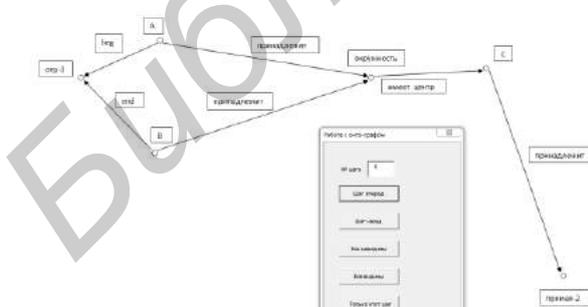


Рисунок 3 – Онтологическая структура после шага 5 (отрезок АВ).

Онтологическая структура, соответствующая промежуточному шагу решения задачи представлена на рис. 3.

Полная концептуальная структура решения этой задачи приведена на рис. 4, где А, В, М, С -

экземпляры концепта *точка*; отр-1,2,3,4 - экземпляры концепта *отрезок*; beg и end - отношения между точкой и отрезком - *является началом* и *является концом*.

Эта структура соответствует онтологической визуализации. С помощью специализированных базовых операций администратор онтологии может проследить по шагам процесс формирования структуры и эвристические основания при выборе каждого шага. Средства фильтрации онтологии позволяют выделить нужный фрагмент структуры и существенно уменьшить громоздкость визуализации.

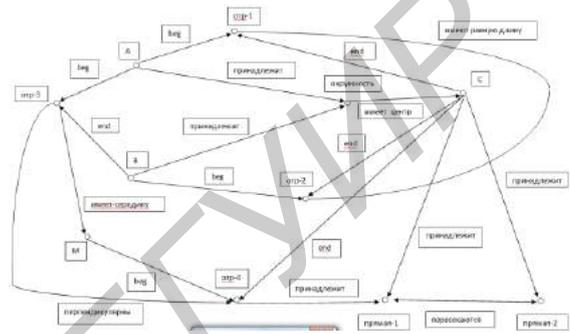


Рисунок 4 – Концептуальная структура решения

Предметная визуализация представляет собой динамически формируемый чертеж, соответствующий выполнению текущей базовой операции. Для нашей задачи базовыми операциями, приводящими к решению, являются:

- построить отрезок А-В;
- найти середину отрезка А-В;
- построить прямую, перпендикулярную отрезку А-В и проходящую через его середину (прямая-2);
- найти точку пересечения прямых прямая-2 и прямая-1 (точка-С);
- построить окружность с центром в точке (точка-С) и радиусом (отрезок А- точка-С).

В процессе применения всех базовых операций найденного решения чертеж модифицируется. Графический результат, соответствующий решению, может иметь вид, приведенный на рис. 5. Именно этот чертеж с комментариями выдается после автоматического нахождения решения.

В дополнение к визуализации вместе с чертежом выводится файл с протоколом выполнения базовых операций. Этот файл имеет вид, практически совпадающий с ЕЯ-описаниями, но возможно дополненный координатами точек, длинами отрезков и т.д., а также эвристическими комментариями по выбору базовой операции.

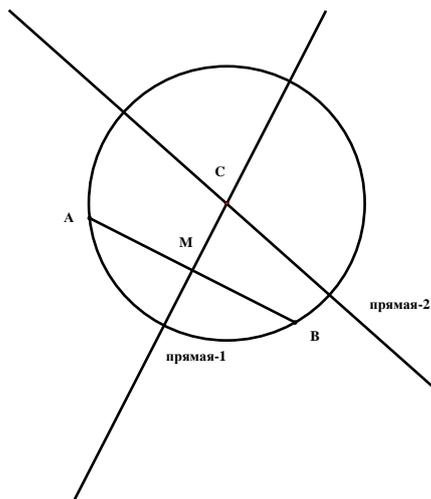


Рисунок 5 – Графический вид решения задачи на построение.

Заключение

В данной работе описывается программное обеспечение, использующее онтологию:

- при лингвистической трансляции условий задачи;
- при автоматическом поиске онтологически-ориентированного решения задачи;
- при демонстрации решения на графическом уровне: онтологический граф и визуальное представление графическими объектами.

Эксперименты подтвердили работоспособность разработанного программного обеспечения, ряд наиболее интересных технических результатов приведен в [Лобзин, 2015].

В прикладном аспекте данное ПО целесообразно развивать для его использования в образовательном процессе. Компьютерная реализация намеченной Пойя концепции [Пойя, 1970] обещает существенно улучшить качество обучения. Эта концепция предполагает формулировку учеником правдоподобных рассуждений и выдвижение догадок и именно на этот процесс сделан акцент в реализованном ПО.

Работа выполнена при поддержке гранта РФФИ 15-07-03847 А «Интерпретация технически-ориентированных текстов на естественном языке с помощью прикладных онтологий».

Библиографический список

- [Khakhalin at al., 2012] Khakhalin G., Kurbatov S., Naidenova K., Lobzin A. Integration of the Image and NL-text Analysis/Synthesis Systems. In book: "Intelligent Data Analysis for Real-Life Applications: Theory and Practice" (Ed. Rafael Magdalena at all). – USA: IGI Global, 2012, p. 160-185.
- [Курбатов и др., 2015] Курбатов С.С., Лобзин А.П., Хахалин Г.К., Лингвистическая поддержка системы решения задач по планиметрии // XX международная научно-практическая конференция "Инновации в науке: применение и результаты". Новосибирск: M&S, 2015, с. 17-23.
- [Апресян и др., 1989] Апресян Ю.Д., Лингвистическое обеспечение системы ЭТАП-2 / Ю.Д. Апресян, И.М. Богуславский, Л.Л. Иомдин и др. – М.: Наука, 1989.

[Давыденко и др., 2011] Давыденко И.Т., Интеллектуальная справочная система по геометрии, OSTIS-2011, / И.Т. Давыденко [и др.]; – Минск. БГУИР, 2011.

[Курбатов, 2010] Курбатов С.С., Высокоуровневые эвристики для автоматизированного формирования базы знаний, / С.С. Курбатов С.С., // 12 национальная конференция по искусственному интеллекту с международным участием КИИ-2010, Труды конференции, М.: Физматлит, Том 2, с. 231-239.

[Геленгер, 1967] Геленгер Г., Реализация машины, доказывающей геометрические теоремы, Г. Геленгер, // Вычислительные машины и мышление, - М.: Мир, 1967, с. 145-165.

[Пойя, 1970] Пойя Д., Математическое открытие, Д. Пойя, - М.: Наука, 1970.

[Лобзин, 2015] http://www.eia--dostup.ru/onto_geom.htm.

[Литвинович, 2012], Литвинович А.В., Язык описания графических объектов GRASP, Литвинович А.В., // Нейрокомпьютеры / №10 за 2012 г., - М.: Радиотехника.

SOFTWARE FOR THE AUTOMATIC SOLUTION OF TASKS ON PLANE GEOMETRY

Curbatow S.S., Lobzin A.P., Khakhalin G.K.

*Research Centre of Electronic Computing
Engineering, Moscow, Russia*

curbatow.serg@yandex.ru

lobzin@rambler.ru

gkhakhalin@yandex.ru

The article describes a system for automatic solution of tasks on plane geometry with the use of ontologies and natural language interfaces

Introduction

This article describes software for automatic solution of tasks on plane geometry with the use of ontologies and natural language interfaces. Integration of linguistic methods, capabilities of the ontology to represent domain knowledge and modern graphics give a fresh look at the role of geometry in research on AI and computer learning.

Main Part

General information about the used ontology tool is provided. The concepts and relationships of the subject area (geometry) are described. For example, specific tasks are demonstrated: the interface with the ontology, the capabilities of the solver. Visualization at the subject and ontological levels is developed for the search solution.

Conclusion

Experiments have confirmed efficiency of the developed software. In applied aspect the software should be developed for use in the educational process.