

УДК 004.414.28

СПРАВОЧНИК ИНТЕГРИРОВАННОЙ СРЕДЫ ИЗУЧЕНИЯ КУРСА АНАЛИТИЧЕСКАЯ ГЕОМЕТРИЯ ДЛЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

**Яцюта В.А., Песчаненко В.С., Львов М.С.
Херсонский государственный университет**

В настоящей статье рассмотрены структура и технологии реализации программного модуля «Справочник» программного средства учебного назначения «Интегрированная среда изучения курса «Аналитическая Геометрия» для высших учебных заведений, которая была разработана по договору с МОН Украины в рамках государственной программы «Информационно-коммуникационные технологии в образовании и науке» в 2007-2008 году. Рассмотрена архитектура и технологии реализации как программного модуля в целом, так и справок как отдельных объектов. Особое внимание уделено алгоритмам выполнения справок, которые по существу играют роль команд пользователя ПМ, осуществляющих символьные преобразования – шаги решения учебной задачи по аналитической геометрии.

***Ключевые слова:** Программные системы учебного назначения, аналитическая геометрия, символьные преобразования, система алгебраического программирования APS, Aplan, математические объекты, математическое ядро.*

Введение

Средство учебного назначения «Интегрированная среда изучения курса «Аналитическая Геометрия» для высших учебных заведений», является одним из серии программных продуктов: («Терм 7-9»[1], «Математическая логика» [2], ...), разработанных НИИ информационных технологий, лабораторией разработки и внедрения педагогических программных средств [3]. Автором идеи этой серии, является Львов М.С., именно им были предложены основные концепции по разработке этой линии программных продуктов.

Ключевую роль при программировании этих учебных средств, сыграла система алгебраического программирования APS[4]. Первая версия системы была разработана в институте Кибернетики им. В.М. Глушкова НАН Украины. Идея создания системы принадлежит доктору физико-математических наук, профессору, академику Летичевскому А.А. После появления первой версии APS была частично переработана и улучшена, кандидатом физико-математических наук Песчаненко В.С. В результате чего были созданы вторая и третья версия системы, на основе которых были разработаны выше упомянутые программные продукты [1,2].

Структура справочника

Составные части

ПМ «Справочник» представляет собой набор методов (функций) реализованных средствами системы алгебраического программирования APS [5] – в языке Aplan (язык этой системы). ПМ предназначен для поддержки пошагового решения задач из курса аналитической геометрии в высших учебных заведениях. Содержание справочника представляет собой набор команд (справок), каждая из которых предназначена для преобразования алгебраического выражения, представляющего учебную задачу. Справочник состоит из двух основных разделов: раздел алгебраических преобразований (равносильные алгебраические преобразования) и раздел геометрических преобразований. Каждый из этих разделов, в свою очередь, структурирован на подразделы, в зависимости от типов объектов над которыми происходят преобразования и элементарных задач. Структура справочника реализована в виде html-документов. Каждая такой документ описывает набор справок, относящихся к тому или иному типу преобразований объектов. Более подробно структура справочника и классификация справок описаны в [6].

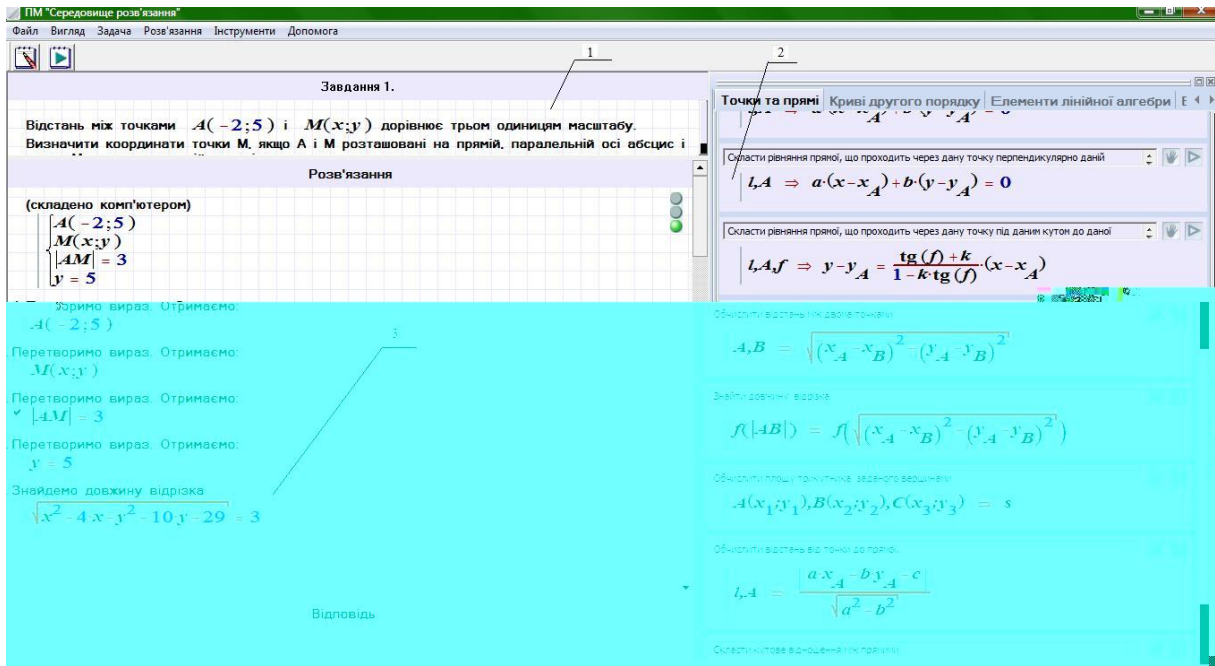


Рис 1. Среда решения(1 – условие задачи, 2 – справочник, 3 – ход решения)



Рис 1. Структура справочника

Структура справок представлена ниже

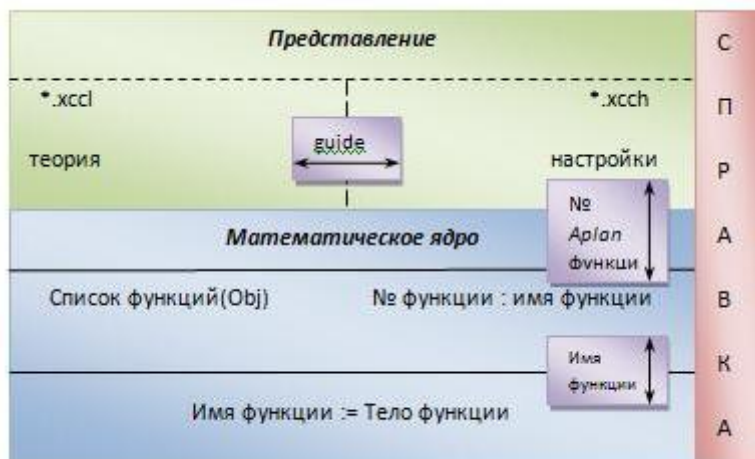


Рис 2. Структура справки

Каждая справка представляет собой объект, состоящий из 2-х слоев: представления (слой интерфейса пользователя), математическое ядро (модуль, выполняющий функции выполнения, хранения алгоритмов написанных на языке Aplan).

Представление.

Этот промежуточный слой отвечает за взаимодействие пользователя с ПМ «Среда решения» – программным модуль «Интегрированной среды изучения курса «Аналитическая Геометрия» для высших учебных заведений, отвечающий за правильность и сохранность хода решения задачи. Основной составляющей «Среды решения», является «Справочник». Слой представляет собой набор XML-документов, в которых описаны функциональности каждой данной справки. Все справки разделены на определенные группы, в зависимости от объектов, над которыми происходят преобразования (числа, уравнения, неравенства, точки и прямые и т.д. (рис. 2)). Поскольку программа поддерживает многоязычность, появилась необходимость отделить текстовую информацию от формульной в справке. В связи с этим каждая группа справок разделена на 2 XML файла, объединенных между собой с помощью уникальных ключей, называемых guide.xscl файл (информация о справке) содержит:

1. Имя справки, характеризующее ее предназначение.
2. Текст, который отображается в поле Среды решения после правильного выполнения справки, перед ее формульным результатом.
3. Информация о том, как выполнить данную справку.
4. Математическое объяснение по данной справке.

xsch файл (настройки) содержит:

1. Номер (ID) Aplan функции, который связывает представление справки с самой справкой в математическом ядре
2. Формула (формулы), что видит пользователь, и которые характеризуют справку
3. Настройки (1 – да, 0 – нет)
 - Будет ли пользователь вводить какие-то данные перед выполнением справки (например, при элементарном преобразовании матрицы нужно передавать коэффициент преобразования, при появлении новой точки (найти середину отрезка и т.д.) нужно вводить имя точки и т.д.)
 - Передавать ли в справку, кроме выделенной части выражения, выражение целиком (Необходимо для разложения определителя матрицы по формуле, преобразования перестановки строк (столбцов) матрицы и т.д.)
 - Нужно ли пользователю выбирать одну из формул (если формул несколько)

Именно набор этих XML файлов и составляет собой справочник.

Математическое ядро

Этот слой отвечает за все математические преобразования. Он полностью написан на языке Aplan программной системы APS. Математическое ядро представляет собой набор функций, разделенных по файлам в зависимости от аргументов, над которыми происходят преобразования. Существует один корневой файл который подключает к себе все остальные, в нем содержится структура (obj), она представляет из себя своеобразный словарь (№ справки (именно по нему идет связь с «представлением», через xsch файл): имя функции (рис. 3)), где каждый объект имеет свой уникальный ключ. Ключом в данном случае является номер функции, а в качестве объекта выступает имя функции. Именно на основе этого файла строится клубок. Клубок – это файл с расширением *.clw, который генерируется средой APS для последующего его использования в программных продуктах, этот файл является физическим представлением математического ядра.

Использование справок

Каждая справка определяет некоторое символьное преобразование данных. В этом преобразовании можно выделить аргументы и результат. Каждую справку характеризует формула (или множество формул), которая определяется следующим образом

$$H_1, \dots, H_k \Rightarrow G \text{ или } H_1, \dots, H_k \Leftrightarrow G$$

H_i – аргументы, G – результат. Справка может быть как однонаправленной, так и двунаправленной. Это означает, что результат может стать аргументом при следующем ее

вызове (в этом случае справка проводит обратные преобразования). Аргументом справки может быть формула, часть формулы, множество формул или объект, имеющий имя. В последнем случае вместо самого объекта можно использовать его имя, поскольку имя объекта является его идентификатором, уникальным ключом в словаре всех объектов. Словарь реализован с использованием библиотеки STL как STL-отображение, интегрированное в систему APS. Этот словарь существует на протяжении жизни клубка, а после начала новой задачи он очищается). Имя каждого объекта определено в строгом синтаксисе, согласованном с требованиями предметной области:

- Точки – одна заглавная буква (либо заглавная буква с индексом)
- Прямая – одна строчная буква (либо строчная буква с индексом)
- Матрица – одна заглавная буква (либо заглавная буква с индексом)
- Определитель – одна заглавная буква (либо заглавная буква с индексом)
- Вектор – одна строчная буква, либо две заглавные (заглавные с индексом) (координаты начала и конца вектора)
- Направленный отрезок – две заглавные буквы (заглавные с индексом) (координаты начала и конца отрезка)

В формулах допускается использование и передача имен. Например, выражение:

$$A \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + B \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \Rightarrow C \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{pmatrix}$$

допустимо заменить на:

$$A + B \Rightarrow C$$

Это существенно сокращает запись выражения и облегчает его восприятие пользователем.

Определенный тип справок, рассчитанных на выполнения некоторой бинарной операции (например, сумма матриц), может в качестве аргументов принимать как выражение целиком, (например, передача **A+B** необходима, когда мы работаем с одной формулой), так и частей выражения (**A, B** когда **A** и **B** – суть операнды отдельных выражений). Такая возможность существенно расширяет функциональность справочника и облегчает решения пользователем задач.

Алгоритм выполнения справки

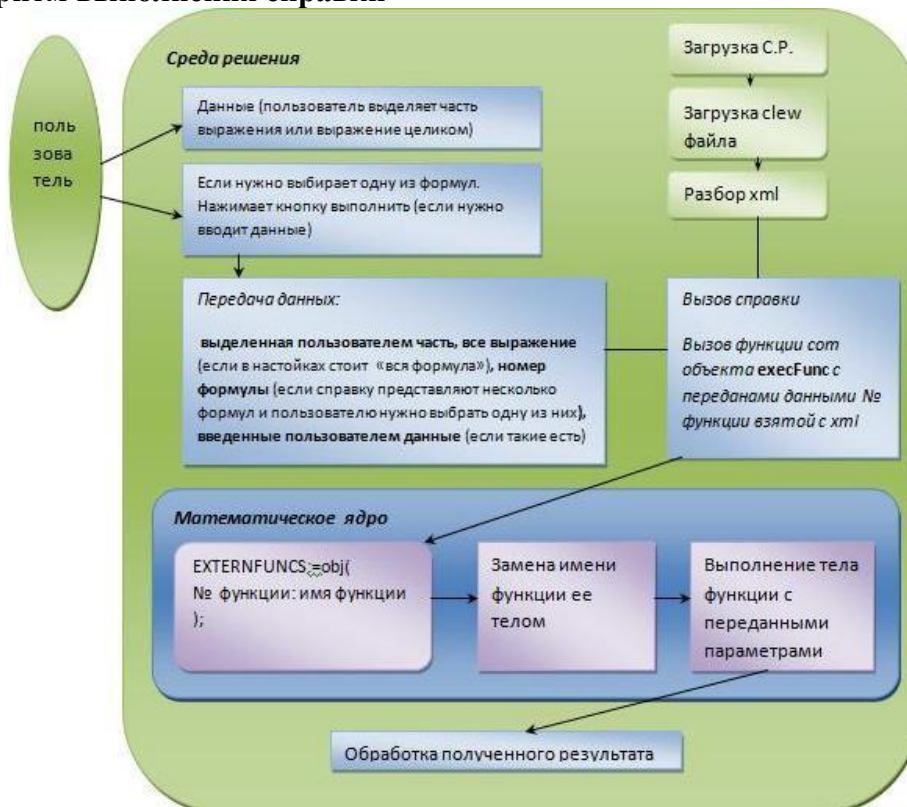


Рис 3. Выполнение справки

При загрузке ПМ «Среда решения» создается экземпляр (назовем его `_aps`) класса `CApsExecutorClass` или `CApsComponentClass`, в зависимости от подключенного `com` объекта.

Далее происходит вызов метода `Construct` объекта `_aps`, в качестве аргумента ему передается абсолютный путь библиотеки `apasm.dll`.

После этого математическое ядро инициализируется методом `InitClew`, в качестве аргумента ему передается строка вида `"-l ap/basis.clew"`, где `l` означает «загрузить», а вторая часть – относительный (к `apasm.dll`) путь к математическому ядру.

Далее происходит обработка `xml` файлов и создается графическое представление справочника.

После загрузки ПМ «Среда решения» пользователь решает определенную учебную задачу, составленную им самим или выбранную из ПМ «Задачник» (ПМ «Задачник» хранит заранее составленные задачи). Решение происходит следующим образом: пользователь выделяет несколько выражений, выражение или его часть затем вызывает справку, которая преобразует формулу (формулы) соответственно своему назначению. Процесс применения справки повторяется до тех пор, пока полученное пользователем выражение не станет ответом задачи.

Рассмотрим отдельно процесс выполнения справки.

После выделения выражения или нескольких выражений пользователь выбирает в справочнике нужную ему справку.

Справка может представляться как одной формулой, так и несколькими (например решить квадратное уравнение при $D>0$, $D<0$, $D=0$), во втором случае на пользователя ложится ответственность за выбор формулы которая характеризует нужное ему преобразование.

Далее пользователь нажимает кнопку «выполнить» Иногда, если нужно передать еще что-то (отношение в для деления отрезка и т.д.), появляется окно с ввода дополнительных данных.

После ввода данных, формируется строка – аргумент которой будет передаваться в `Arlan`-функцию. Строка строится следующим образом (рис. 4): «выделенная пользователем часть» + «, \rangle » + «вся формула» (если в «настойках» стоит «вся формула») + «, \rangle » + «номер формулы» (если справку представляют несколько формул и пользователю нужно выбрать одну из них) + «, \rangle » + «введенные пользователем данные» (если такие есть).

Из `xml`-файла выбирается номер справки и вызывается метод объекта `_aps` `execFunc(arg1, arg2, arg3)`, где `arg1` – строка упомянутая выше, `arg2` – номер `Arlan`-функции (взятый из `xsch` файла), `arg3` – результат выполнения функции (этот аргумент передается по ссылке).

Эта функция вызывает клубок, который обрабатывается специальными `dll` файлами. Клубок строится на основе `*.ar` файла, в котором хранится код программы на языке `Arlan`.

Этот файл содержит в себе набор функций, выполняющих соответствующие преобразования, специальной структуры `EXTERNFUNCS`, которая является словарем, (каждая запись состоит из двух полей ключа – номер функции, объекта соответствующему ключу – имя функции).

На следующем шаге происходит поиск в `EXTERNFUNCS` функции по номеру (`arg2`), после чего по имени функции происходит вызов ее тела с аргументом `arg1`.

Далее аргумент `arg1` по отметкам объектов, описываемых именами, арностями и приоритетами, (например `+, -, &...`), преобразуется в дерево.

Выполнение функции реализовано как переписывание и данного дерева, и деревьев, созданных за время жизни функции и клубка. Время жизни функции – от точки входа в функцию, до слова `return`, время жизни клубка – от запуска «Среды решения» до ее закрытия.

После выполнения функции возвращаемый результат записывается в `arg3`.

Выводы

Технология `XML` – очень удобное средство упорядоченного хранения иерархических данных. Она подходит для хранения описаний справок. Отметим положительные стороны технологии `XML`:

- иерархичность данных,
- удобный доступ к разным частям данных,
- легкость добавления, удаления и изменения записей.

Среди негативных сторон можно отметить:

- сложность изменения структуры хранения данных
- сложность нахождения ошибок в описании справок (порой это приводит к неработоспособности целого раздела)

Как было упомянуто выше, математическое ядро написано на языке Arlan системы алгебраического программирования APS. APS является удобным средством интерпретации алгоритмов, которые построены на преобразования символических выражений. Если же алгоритмы предназначены для численных расчетов, лучше использовать C++, C#. Особенностью APS является использование технологии переписываний, реализованных в виде «переписывающей» машины. Программист задает набор правил (систему правил переписывания) вида: *шаблон = выражение*, где шаблон – дерево определенное пользователем, а выражение – формула, на которую заменяются подвыражения, которые определяемые шаблоном. Программист может использовать либо стандартную, или собственную стратегию обхода дерева выражения и применения к его поддеревьям системы правил переписывания.

Ниже приведена сравнительная характеристика интерпретации алгоритмов на APS и других платформах.

Таблица №1

Сравнительная характеристика времени написания программ

Алгоритм	Время затраченное на написания кода (мин.)		Количество строк кода	
	APS	C++, JAVA	APS	C++, JAVA
Вычисление определителя матрицы	50	30	20	20
Разложение определителя	50	80	25	40
Нахождение канонической формы кривой 2-го порядка	600	800	478	800
Сложение матриц	30	10	26	13

Легко видеть, что APS существенно выигрывает, на классе задач, связанных с преобразованием выражения, а не с вычисляем их значений. Например, задача вычисления определителя является задачей императивного программирования. Наоборот, задача нахождения канонической формы кривой – это задача преобразования формулы. Программы на C-подобных языках на этот процесс тратят достаточно много времени, поскольку необходима разработка парсера выражения. На APS это это делается достаточно легко. Т.о. использование АПС для указанного класса существенно облегчает работу программиста.

При реализации алгоритма сложения матриц проиграла не система APS а принятое архитектурное решение, матрицы пришлось представлять в виде строк типа 'Matrix(A,r(1,2,3),r(4,5,6),r(4,5,6))'. Поэтому пришлось писать функцию преобразования строки в динамический массив.

Положительные стороны использования APS:

- Легкость изучения языка.
- Быстрая реализация алгоритмов.
- Код легко понимать программисту, который видит его впервые.

- Возможность изменения кода программы непосредственно во время ее выполнения.
- Возможность использования стандартных и «своих» стратегий обходов деревьев.
- Система достаточно устойчива, даже при нахождении ошибок продолжает свою работу

К отрицательным сторонам можно отнести:

- Легкость допущения таких ошибок, как отсутствие «(», «,», «;»...
- Сложность отладки алгоритмов, заключается в том, что при трассировке системы правил переписывания приходится просматривать каждую попытку применения правила.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. [<http://www.kspu.edu/About/Downloads/LabRVPPZ/Term.htm>].
2. [<http://www.kspu.edu/About/Downloads/LabRVPPZ/Term.htm>].
3. [<http://kspu.edu/about/Institutes.aspx?12=96>]
4. Песчаненко В.С. Розширення стандартних модулів системи алгебраїчного програмування APS для використання у системах навчального призначення // Науковий часопис НПУ імені М.П. Драгоманова Серія №2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. Пр./Редкол. – К.:НПУ ім.М.П.Драгоманова, – №3 (10), 2005. – С.206-215.
5. Львов М.С. Основні принципи побудови педагогічних програмних засобів підтримки практичних занять Управляючі системи і машини. 1: 68-74. (2008).
6. Львов М.С. Проектування логічного виводу як покрокового розв'язання задач в математичних системах навчального призначення. Управляючі системи і машини. – 6:(70-75 (2006).
7. Львов М.С., Співаковський О.В. Методи проектування систем комп'ютерної підтримки математичної освіти Математичні моделі і сучасні інформаційні технології: Зб. наук. пр./НАН України. – Київ, 1998. – С.101-111
8. Песчаненко В.С. Використання системи алгебраїчного програмування АПС для побудови систем підтримки викладання алгебри у школі. Управляючі системи і машини. 4: 86-94 (2006)
9. Песчаненко В.С. Алгоритми підтримки кроку розв'язання алгебраїчних задач у шкільній системі комп'ютерної алгебри ТерМ. Управляючі системи і машини. 1: 61-68 (2007)