

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Томский государственный университет
Горно-Алтайский государственный университет
Институт оптики атмосферы им. В.Е. Зуева СО РАН

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ИССЛЕДОВАНИИ СЛОЖНЫХ СТРУКТУР

**МАТЕРИАЛЫ ДЕСЯТОЙ РОССИЙСКОЙ КОНФЕРЕНЦИИ
С МЕЖДУНАРОДНЫМ УЧАСТИЕМ**

Томск
Издательский Дом Томского государственного университета
2014

фичными требованиями, например, требованиями к безопасности для возможности приема модулей сторонних разработчиков.

Для поддержки горизонтального масштабирования система разделена на несколько узлов, часть из которых не являются масштабируемыми (например, корневой узел системы, отвечающий за взаимодействие остальных узлов), часть – масштабируемыми (например, проектный узел, отвечающий за непосредственное редактирование пользователями диаграмм).

С точки зрения серверной части, поддержка разнородных клиентских приложения затруднена различными возможностями по передачи сообщений (протокол, используемый для передачи и способ кодирования сообщений). Для решения этой проблемы разработан механизм расширения поддерживаемых протоколов и способов кодирования, который позволяет без внесения модификаций в саму систему добавлять поддержку необходимых протоколов или способов кодирования.

Поддержка необходимых возможностей объектной моделью реализована с помощью специализированных свойств и коллекций, которые с помощью локального контекста потока исполнения ведут учет всех изменений и позволяют: выполнять откат отдельных действий, производимых над объектами, осуществлять проверку наличия у пользователя необходимых прав, а также обращаться к предыдущим версиям объектов.

Расширение системы осуществляется путем динамического подключения специально оформленных библиотек, содержащих классы и различные ресурсы, описывающие элементы, поддержку которых необходимо добавить. Для защиты системы и ее окружения эти библиотеки загружаются в отдельный домен (AppDomain), имеющий минимально возможный набор прав, необходимый для исполнения кода подключенных библиотек. Для ограничения ресурсов, потребляемых тем или иным модулем расширения планируется реализация специального CLR-хоста с возможностью реализации соответствующих ограничений.

В зависимости от окружения в котором выполняется серверная часть системы («облачная» платформа, корпоративная сеть, отдельный сервер) могут быть доступны те или иные способы хранения информации (файлы, СУБД, «облачные» СУБД и хранилища файлов). Для предоставления возможности выбора соответствующей системы хранения реализована возможность подключения дополнительных модулей для работы с соответствующими системами хранения.

Литература

1. *Моисеев А.Н., Политов А.М., Хомич М.О.* Концепция системы поддержки командной разработки программного обеспечения с синхронизацией в режиме реального времени // Информационные технологии и математическое моделирование (ИТММ-2011) : матер. X Всерос. науч.-практ. конф. с междунар. участием (25–26 ноября 2011 г.). Томск, 2011. Ч. I. С. 69–71.

ПРИМЕНЕНИЕ ГРАФИЧЕСКОЙ БИБЛИОТЕКИ GDI+ ДЛЯ РЕШЕНИЯ ЗАДАЧ АППРОКСИМАЦИИ И ИНТЕРПОЛЯЦИИ

А.В. Приступа

Национальный исследовательский Томский государственный университет, Томск, Россия
pristupa@sibmail.com

Пусть имеется набор точек (X_i, Y_i) , $i = \overline{1, m}$, для которых $X_{i+1} > X_i$ для $\forall i \in [1, m-1]$. Одной из задач, часто возникающих на практике, является задача интерполяции, которая состоит в нахождении коэффициентов сплайна, проходящего через все эти точки. Для точного ее решения необходимо составить систему линейных уравнений на основе условий интерполяции, непрерывности и выбранных краевых условий [1]. Цель состоит в том, чтобы с помощью найденных коэффициентов определить значение Y для произвольного X из диапазона $[X_1, X_m]$. При этом поиск коэффициентов сплайна является, как правило, не отдельной самостоятельной задачей, а лишь некоторой подзадачей, выполняемой в рамках более сложного алгоритма. И зачастую исследователю не требуется точное решение, для которого нужно строить систему линейных уравнений, приравнивая нулю значения младших производных (для записи краевых условий) и т.п. Ему достаточно приближенного решения, определенного с требуемой точно-

стью. Такое решение можно получить при помощи стандартной библиотеки GDI+ [2], которая используется в Windows для работы с 2D-графикой.

На начальном этапе необходимо с помощью средств библиотеки построить кривую, соединяющую все точки (X_i, Y_i) . Для этого потребуется создать объект типа *GPGraphicsPath* и вызвать его метод *AddCurve*, основными аргументами которого являются массив координат точек и его размерность. Однако существует бесконечное множество кривых, проходящих через заданные точки. Поэтому у метода *AddCurve* имеются еще дополнительные аргументы. В частности, параметр *Tension* позволяет управлять степенью изогнутости (упругости) сплайна. Значение, равное 0, соответствует бесконечной физической упругости, из-за которой соответствующая кривая соединяет каждые две соседние точки по кратчайшему пути (по прямой). С ростом значения параметра *Tension* от 0 до 1 (по умолчанию используется значение, равное 0,5) сплайн становится все менее упругим (более изогнутым), а при значениях, больших 1, начинает походить на сдавленный берегами ручей, стремящийся увеличить изгиб и течь по более длинному пути (рис. 1).

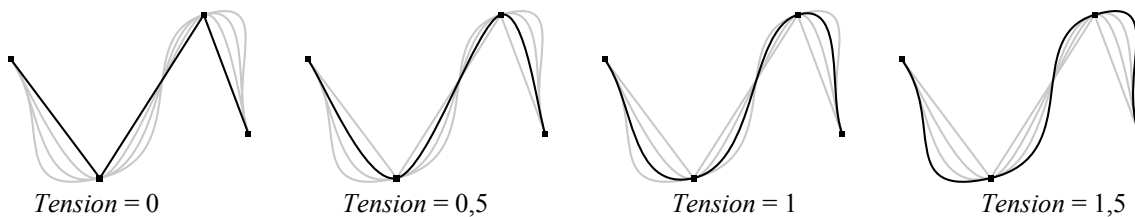


Рис. 1. Сплайны с различным значением параметра *Tension*

Кроме того, поскольку мы явно не задаем краевых условий, то для уточнения формы крайних сегментов можно воспользоваться разновидностью метода *AddCurve* с дополнительными параметрами *Offset* и *NumberOfSegments*, передав в этот метод массив с большим количеством точек, чем требуется. Параметр *Offset* определяет индекс точки, с которой начнется построение кривой, а параметр *NumberOfSegments* указывает количество сегментов. Остальные (невидимые) точки будут использованы для расчетов.

При желании можно вызвать метод *Draw* объекта *GPGraphicsPath* и отобразить на экране полученную кривую, но можно этого и не делать. Далее нам требуется определить неизвестное значение Y по известному значению аргумента X_0 . Такой встроенной функции у GDI+ нет. Однако у объектов *GPGraphicsPath* имеется замечательный метод *Flatten*, позволяющий аппроксимировать контур отрезками ломаной. Точность аппроксимации определяется аргументом *Flatness*. После завершения работы метода нам остается с помощью *GetPathPoints* получить координаты точек ломаной.

Имея массив координат точек, найдем сегмент ломаной, содержащий значение X_0 , и определим точку пересечения найденного сегмента с вертикальной прямой $x = X_0$. Ордината этой точки и будет являться приближенным решением с точностью, которая регулируется значением аргумента *Flatness* метода *Flatten*. Таким образом, нам удалось решить математическую задачу с помощью графической библиотеки, затратив на это гораздо меньше усилий по сравнению с поиском аналитического решения.

Литература

1. Приступа А.В. Компьютерная графика. Алгоритмические основы и базовые технологии : учеб. пособие. Томск : Изд-во НТЛ, 2012. 260 с.
2. MSDN Library: Using GDI+. URL: [msdn.microsoft.com/en-us/library/ms533802\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533802(v=vs.85).aspx)

ОПЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ КОНВЕЙЕРНОЙ ОБРАБОТКИ НЕОДНОРОДНОГО ПОТОКА

С.П. Сущенко, Р.В. Ткачев

Национальный исследовательский Томский государственный университет, Томск, Россия
ssp@inf.tsu.ru, tkachevrv@mail.ru

Одной из наиболее значимых качественных характеристик обработки неоднородного потока заявок в неоднородном конвейере является общее время обработки всех заявок конвейером [1]. Важнейшим