Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования

Интеграция CASE-системы QReal с математическим пакетом Scilab

Курсовая работа студента 445 группы

Савина Никиты Сергеевича

Научный руководитель ……. Ю.В. Литвинов

Санкт-Петербург

2010

Оглавление

Введение

Активное развитие программной инженерии в настоящее время увеличивает потребность в средствах, делающих процесс разработки более продуктивным. Существует довольно большое количество подобных средств, среди которых мы рассмотрим подробнее CASE-системы. Они могут использоваться на всех этапах жизненного цикла программного обеспечения: от анализа предметной области и формализации требований до проектирования, реализации, генерации кода и отладки. На кафедре системного программирования математико-механического факультета СПбГУ уже несколько лет разрабатывается своя CASE-система QReal. Её важной особенностью является возможность с помощью метамоделирования быстро и удобно реализовывать новые предметно-ориентированные визуальные языки. Для этого QReal предоставляет такие инструменты, как визуальный метаредактор с редактором графического представления элементов модели, возможность автоматической компиляции созданной метамодели языка в плагин для среды QReal, реализующий графический редактор этого языка. Кроме того, QReal предоставляет возможности для написания генераторов исходного кода по визуальным моделям. На данный момент в QReal реализована генерация в Java из моделей на языке UML, генерация в язык описания аппаратных систем HaSCoL и несколько других генераторов.

Очередная задача возникла в результате сотрудничества разработчиков QReal и кафедры теоретической кибернетики. Кафедра занимается такими задачами, как устойчивость и колебания динамических систем, оптимальное и адаптивное управление, управление сложными системами и хаосом, алгоритмы обработки сенсорной информации и теоретическая робототехника, управление и оценивание через коммуникационные сети. В последнее время сферы IT и кибернетики стали еще ближе из-за распространения такого направления научной деятельности, как программное управление роботами. Возможность приобщиться к созданию роботов предоставила компания LEGO, которая выпустила серию конструкторов LEGO Mindstorms, имеющих в комплекте набор стандартных деталей LEGO, а также сенсоров и программируемого блока. Изначально предполагалось, что распространение роботов будет происходить исключительно в игровой индустрии, поэтому штатные средства написания ПО, такие как NXT и RoboLab, не рассчитаны на решение сложных ресурсоемких задач. После того, как управлением роботами помимо детей заинтересовались ученые и инженеры, появилось большое количество продуктов, позволяющих компилировать код на C под процессоры NXT:

Microsoft Robotics, NBC, NXC, RobotC, LabVIEW Education Edition, LeJOS NXJ, pbLua, LEJOS OSEK, ICON. Однако кибернетиком этого оказалось недостаточно, так как была необходимость обрабатывать сигналы от сенсоров в реальном времени, например, для создания робота, балансирующего на шаре. Возможностей встроенной операционной системы от LEGO не хватало в плане скорости работы, поэтому приходилось прибегать к таким решениям, как запуск программы без ОС вообще, или же использование собственной ОС. Кроме того, на кафедре решались специфические задачи, которые требовали сложных математических вычислений.

Необходимо было использовать мощный математический пакет для программирования, после чего из соображений скорости работы генерировать исполняемый код на C. На данный момент для решения задачи используется Matlab. Язык Matlab является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования. Для генерации кода на C используется специальный плагин. Полностью интегрированный с Matlab интерактивный инструмент для моделирования динамических систем Simulink дает возможность строить диаграммы, имитировать динамические системы и исследовать работоспособность систем. Несмотря на большое количество положительных моментов, существенным недостатком Matlab является его высокая стоимость. И если университет мог позволить себе покупку нескольких лицензий, то кружок юных кибернетиков из физико-математической школы №239, который тесно сотрудничает с кафедрой теоретической кибернетики, оказался к подобным тратам не готов. Возникла потребность использовать бесплатные аналоги.

В данном случае выбор не такой большой, так как из бесплатных математических пакетов, которые предоставляют достаточную функциональность, можно назвать только Scilab и предоставляемое им средство визуального моделирования Scicos(XCos). Несмотря на положительные стороны этого продукта, существуют следующие проблемы: Scilab не предоставляет возможности генерировать код на C, а XCos сложен в использовании. Поэтому было принято решение внедрить для решения данной задачи систему QReal, по максимуму переиспользовав Scilab и написав для него графический фронтенд, где можно рисовать диаграммы удобнее, чем в XCos, а также обеспечив кодогенерацию. Таким образом, мы получим для рисования диаграмм научных расчётов преимущества CASE-системы: привязку к другим диаграммам, интеграцию с системами контроля версий,

удобный и специально сделанный для рисования диаграмм пользовательский интерфейс, возможности интеграции с другими языками, а также быструю расширяемость за счет метаязыковых средств QReal.

Целью курсовой работы является переиспользование кода Scilab для проведения математических вычислений по схемам, задаваемым в виде диаграмм в QReal и генерации кода на C, реализующем эти вычисления. Требуется создать в QReal предметно-ориентированный язык для описания математических расчетов и редактор диаграмм, а также реализовать генератор кода на C, осуществляющего вычисления по предоставленной схеме.

Технологии

Scilab

Scilab – пакет прикладных математических программ, предоставляющий открытое окружение для инженерных и научных расчетов. Scilab распространяется вместе с исходным кодом с 1994 года. В 2003 году для поддержки Scilab был создан консорциум Scilab Consortium. Сейчас в его состав входят 25 участников, в том числе Mandriva, INRIA и ENPC. Программа доступна для различных операционных систем, включая Microsoft Windows и GNU/Linux. Возможности Scilab могут быть расширены внешними программами и модулями, написанными на разных языках программирования. Программа имеет открытый исходный код, и её лицензия позволяет свободное коммерческое использование и распространение измененных версий, которые должны включать в себя исходный код. Для коммерческого распространения измененных версий необходимо согласование с INRIA. Начиная с версии 5.0 программа распространяется под совместимой с GNU GPL 2 лицензией CeCILL.

Scilab имеет схожий с Matlab язык программирования. В состав пакета входит утилита, позволяющая конвертировать документы Matlab в Scilab.

Scilab позволяет работать с элементарными и большим числом специальных функций (Бесселя, Неймана, интегральные функции), производить численные вычисления, решать задачи линейной алгебры, оптимизации и симуляции, использовать мощные статистические функции, имеет мощные средства работы с матрицами, полиномами а также средство для построения и работы с графиками.

Для численных расчетов используются библиотеки Lapack, LINPACK, ODEPACK, Atlas и другие.

В системе доступно большое число инструментов:

- 2D и 3D графики, анимация
- линейная алгебра, разреженные матрицы
- полиномиальные и рациональные функции
- интерполяция, аппроксимация
- решение ДУ
- обработка сигналов

- параллельная работа

- статистика

- интерфейс к Fortran, C, C++, Java, LabVIEW

Выделим положительные стороны данного продукта и подведем итог:

- Бесплатность

- Scilab содержит большое количество математических функций

- Есть возможность добавления новых функций, написанных на других языках программирования (C, C++, Fortran)

- Имеются разнообразные структуры данных (списки, полиномы, рациональные функции, линейные системы)

- По синтаксису язык схож с Matlab, имеется возможность конвертации из Matlab в Scilab

- Предоставляет для работы аналог Simulink в пакете Matlab - Scicos (XCos)

- Возможность запуска в консоли без использования графического интерфейса, в том числе в версии под Windows. Это позволяет проводить автоматизированные вычисления.

- Большое количество руководств и мануалов.

В результате анализа данного продукта было принято решение использовать Scilab для нашей задачи. Scilab бесплатен, предоставляет разработчику мощные средства и удобен в использовании.

Scilab2c

Scilab не предоставляет возможность генерации кода на C, поэтому для решения данной задачи была использована технология Scilab2c .

Scilab2c – инструмент, который используется для генерации Scilab кода в код на C. Он был разработан компанией hArtes, деятельность которой направлена на создание нового целостного подхода к разработке сложных гетерогенных встраиваемых решений. Scilab2c является интерфейсом между пакетом Scilab и дизайнером потоков от hArtes.

Полученный в результате работы приложения код имеет типичную для языка C структуру и набор функций, причем никаких отсылок с исходному коду на Scilab не имеет. Другими

словами, сгенерированный код независим в том смысле, что для запуска сгенерированного приложения не требуется интерпретатор Scilab, а количество необходимых C-файлов минимизировано.

QReal

Среда QReal предоставляет возможность быстрого и удобного создания новых редакторов. Это достигается за счет возможностей для реализации новых визуальных языков с помощью визуального метаредактора и последующей компиляции созданной метамодели.

Метамодель должна задавать синтаксические правила языка. Во многих CASE-системах метамодели представляются и хранятся по-разному. В QReal метамодель может представляться в нескольких видах:

- как модель метаредактора. Модель хранится в репозитории QReal как обычная модель визуального языка

- в виде xml файла, который содержит в себе описание графического представления фигур

Чаще используется именно второй вариант. В xml хранится следующая информация об объектах:

- имя диаграммы

- неграфические элементы диаграммы: сейчас к ним относятся только типы-перечисления. Их можно использовать как наборы допустимых значений свойств элементов.

- графические элементы диаграммы (т.е. те, которые доступны из палитры)

    - элементы (node) – узлы графа диаграммы, представляют логические сущности, являющиеся элементами разрабатываемой системы.

        - имя элемента: название, которое будет отображаться на палитре рядом с элементом

        - графика: каждый элемент должен отображаться на сцене. Для него задается размер. Изображение описывается как набор графических примитивов: линий, эллипсов, прямоугольников и т.д. Имеется возможность задавать для

них стили линий (сплошная, пунктирная, и т.д.), цвет, и другие графические свойства

- свойства: неграфические свойства элемента, которые могут заполняться пользователем при построении диаграммы. Имеют имя, тип значения (в QReal поддерживаются следующие типы: строковый, типы-перечисления (включая булевский), типы-ссылки (значение, являющееся ссылкой на другой объект в модели)), значение по умолчанию

- порты: это те места графического изображения элемента, куда присоединяются линии связей. Задаются координатами. В QReal используется два вида портов: точечный и линейный.

- взаимодействия с другими элементами: может ли данный элемент содержать или содержаться в указанном элементе или соединяться с ним с помощью связи.

- связи (edge) - ребра графа диаграммы, представляют связи между сущностями.

- имя связи: задается по аналогии с именем элемента

- графика: задается стиль линии

- свойства: по аналогии со свойствами элемента

В QReal есть дизайнер элементов диаграмм, который позволяет конвертировать полученный результат в xml формат, что значительно сокращает время создания диаграмм и улучшает их внешний вид. Дизайнер элементов позволяет рисовать, как примитивные фигуры, так и сложные кривые, создавать надписи, использовать различные стили, шрифты и пр. Также предусмотрена возможность задания точечных и линейных портов. После создания диаграммы в дизайнере есть возможность сохранить результат в виде картинки или сгенерировать xml файл, содержащий графическую информацию об объекте, который после будет использован в метамодели для описания графики элемента.

Метамодели различных диаграмм обрабатываются генератором редакторов, который по ним генерирует код на C++. Полученный код компилируется вместе с кодом QReal, и получается плагин – dll файл (один плагин может содержать несколько редакторов), удовлетворяющий определенному интерфейсу. Этот плагин представляет классы, реализующие отображение фигур, и умеющие выдавать информацию о фигурах, типа

свойств, портов, допустимых связей и т.д. Приложение использует полученную информацию для того, чтобы реализовывать поведение редактора.

Программная реализация

Генерация кода на С

Для генерации кода на С Scilab2c требуется передать файл scilab (.sci, .sce), поэтому генерация происходит по схеме

QReal – > Scilab – > Scilab2c – > C

1. QReal – > Scilab

   В результате работы приложения QReal по диаграмме генерируется код на Scilab, в соответствии с требованием: набор команд должен представлять собой функцию. Теоретически после этого можно открыть приложение Scilab и запускать scilab2c из него, не используя QReal.

2. Scilab – > Scilab2c

   поскольку Scilab2c - это скрипты на языке Scilab, то чтобы ими воспользоваться, нужно программно запустить Scilab с помощью команды StartScilab и передавать ему команды. В том числе, запустить сам Scilab2c, выполнив loader.sce.

   Для генерации кода с помощью программы Scilab2c ей требуется передать на вход Scilab файл с расширением .sci или .sce., созданный на этапе QReal – > Scilab.

3. Scilab2c – > C

   По окончании работы scilab2c будет сгенерирован код на С, а именно два файла – main.h, main.cpp. Для того чтобы скомпилировать полученный код – необходимо подключить все нужные Scilab2c библиотеки. После этого для запуска приложения библиотеки должны находиться в папке с бинарным файлом.

Эта схема предоставляет возможность после генерации просмотреть scilab файл и убедиться, что все работает корректно, выполнив файл в среде Scilab.

Программа Scilab2c должна быть установлена на машине, на которой производится генерация кода, но получившееся в результате приложение требует только библиотеки Scilab2c , что позволяет использовать результат для работы с роботами, память которых ограничена.

Создание математического редактора

С помощью средств, предоставляемых риалом, был создан новый математический редактор. Для решения данной конкретной задачи были необходимы некоторые усовершенствования, без которых работа с математическими моделями была бы затруднена. Рассмотрим их подробнее.

Именованные порты

В QReal порты используются для взаимодействия элементов диаграмм. По каждому порту можно запросить информацию о связи, которая входит или выходит из данного порта, а затем соответственно информацию о связанном объекте. При создании математического редактора возникла потребность каким-то образом идентифицировать порты. Рассмотрим это на примере элемента интеграла

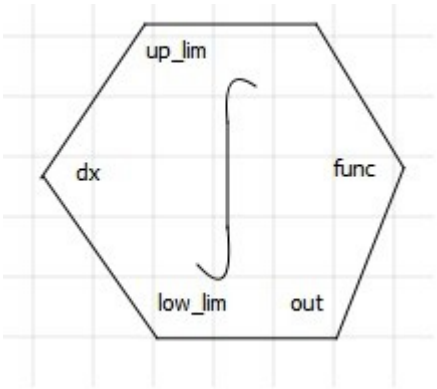

Рис. 1

В примитивном случае для задания интеграла требуется указать следующие значения: верхний предел, нижний предел, подынтегральная функция, переменная интегрирования. Пользователю предлагается присоединить к портам элементы диаграмм, значения которых соответствуют подсказкам, написанным около портов

Для того чтобы проверять синтаксическую корректность диаграммы прямо в процессе её рисования, нужно уметь задавать в метамодели, к какому именно порту какие связи и элементы могут быть подключены, однако на данный момент QReal не обладает такими возможностями.

При генерации кода по диаграмме мы должны получить необходимую информацию именно с того порта, который нам нужен. К сожалению, QReal не предоставлял такой возможности, так как для большинства существующих визуальных языков точки присоединения связей не несут дополнительной семантической нагрузки. Поэтому пришлось расширить язык описания метамоделей, добавив в него возможность задавать информацию об имени порта. Теперь для каждого порта фигуры можно указать его имя в

виде строки, после чего при генерации кода получать присоединённую к этому порту связь или элемент на другом конце связи, обращаясь к порту по его имени.

Свойства элемента или внешние связи?

В результате реализации именованных портов мы получили возможность задавать необходимые свойства с помощью внешних элементов. Рассмотрим диаграмму интеграла, созданную в соответствии с новой концепцией.

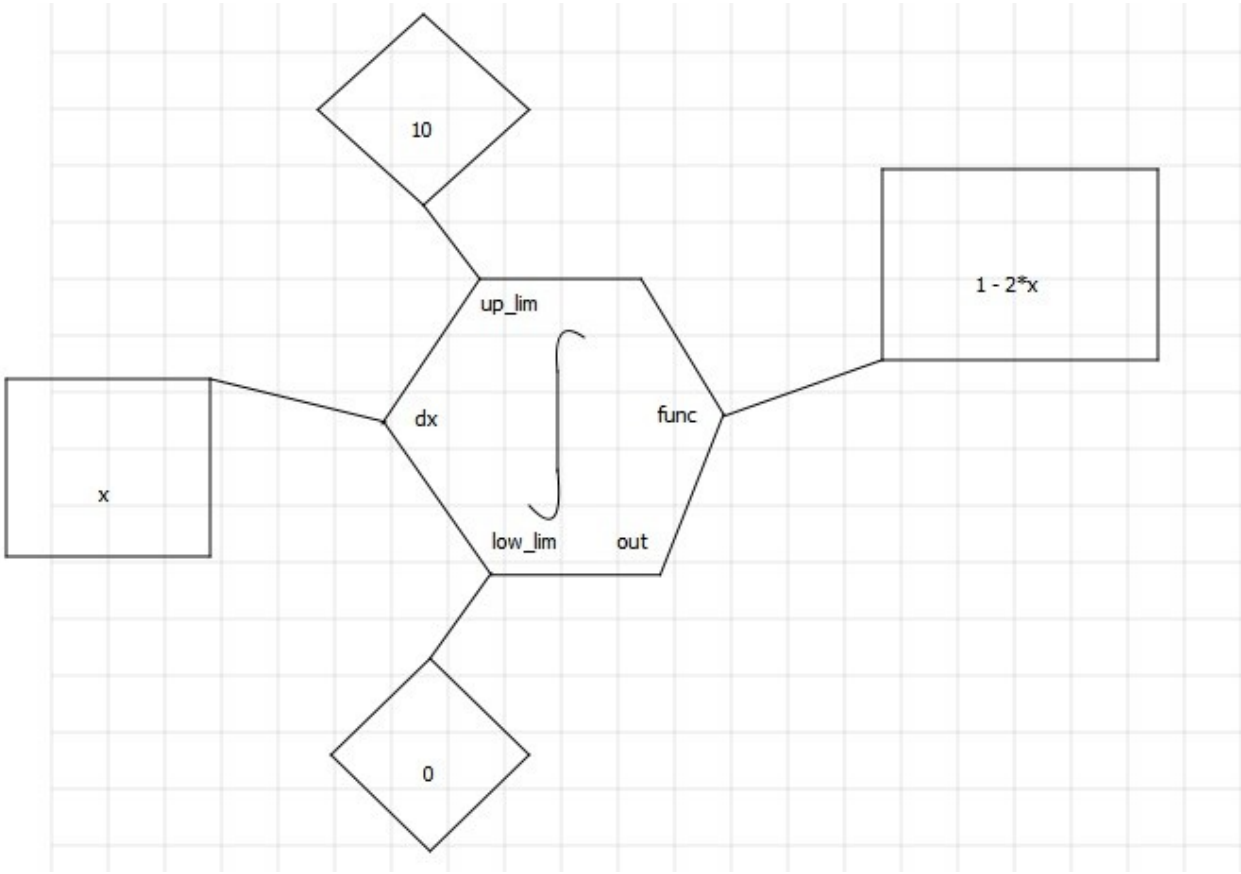

Рис. 2

По данной диаграмме сгенерируется код, который будет считать значение интеграла

$$\int_0^{10}(1-2x)dx$$

Однако получившаяся диаграмма занимает много места и, следовательно, сложна для восприятия. Такой вариант разумно использовать тогда, когда в качестве входящего значения для данного элемента используется исходящее значение другого элемента. В противном случае, достаточно определить значение в качестве свойства элемента

интеграла. Было принято решение дать пользователю возможность задавать необходимые свойства, как с помощью внешних элементов, так и с помощью свойств. Таким образом, была решена проблема задания значений по умолчанию, так как в QReal уже была реализована функциональность, отвечающая за дефолтные значения свойств.

Был установлен следующий приоритет значений:

1. Внешние элементы
2. Свойства
3. Значения по умолчанию

В результате диаграмма выглядит компактнее, а пользователь избавлен от необходимости указывать необходимое значение, если его устраивает дефолтное. В дальнейшем предполагается, что пользователь сможет настраивать значения по умолчанию.

Семантика языка

Для построения диаграмм используются блоки и связи. С блоком связан набор входящих значений и исходящее значение, которые могут быть какими-то из следующих типов:

- число (+- ∞ считается числом)

- функция

- вектор

- матрица

Между этими типами существуют отношения включения: число является частным случаем функции, а вектор – частным случаем матрицы.

Входящие значения для блока могут быть представлены в виде внешних блоков и связей между соответствующими портами, а также в виде свойств данного блока. Связи могут быть направленными и ненаправленными. Первые употребляются в том случае, если необходимо связать два вычисляемых блока, тем самым задав порядок вычисления. Ненаправленные используются для входящих константных значений. На входящие значения накладываются ограничения по типу, при этом сохраняется отношение

включения (т.е. если для блока и конкретного порта входящее значение должно быть типа «функция», то программа отработает корректно и в том случае, если входящее значение типа «число»).

Каждый блок преобразует входящие значения в результат заданного типа согласно правилу, соответствующему данному блоку. После чего исходящее значение может быть использовано в качестве входящего для другого блока.

Вычисление производится, начиная с блоков без входов (направленных связей, входящих в данный блок) в порядке обхода в ширину. Вычисление следующего блока не может начаться до тех пор, пока не вычислены все исходные значения.

Алгоритм обхода дерева вычислений реализован следующим образом:

Ненаправленные связи алгоритмом игнорируются. Производится поиск элементов, не имеющих входящих направленных связей. Эти блоки могут быть посчитаны, так как имеются все необходимые значения. Затем из участвующих в поиске блоков и связей будут исключены найденные блоки и исходящие из них связи. Снова будет произведен поиск элементов, не имеющих входящих направленных связей. Найденные блоки и связи, исходящие из них, убираются из дерева поиска. Так происходит до тех пор, пока не будут вычислены все блоки.

В результате обхода всех блоков мы получаем значение последних блоков, которые не имеет выходной связи. Эти значения будут выданы в результате работы сгенерированной программы на C.

Пример использования

Пусть пользователю необходимо провести вычисления по формуле, приведённой на рис. 3

$$\left(\int_{\frac{\pi}{2}}^{\pi} cosx\,dx + \int_{\pi}^{\frac{3}{2}\pi} sinx\,dx\right)^{10}$$

Рис. 3

Построим диаграмму, соответствующую данному примеру. Диаграмма будет состоять из двух блоков интеграла, блока суммы и блока степени. Для интегралов все необходимые значения можно задать в виде свойств, так как они константные. Блок суммы должен получить в качестве входящих значений результаты выполнения блоков интегралов, поэтому оба блока интеграла должны быть соединены с блоком суммы направленной связью. Блок степени должен получить в качестве входящих значений основание степени и показатель степени. В качестве основания степени используется результат вычисления блока суммы, поэтому блок суммы должен быть соединен направленной связью с блоком степени, а показатель степени может быть задан свойством блока степени.

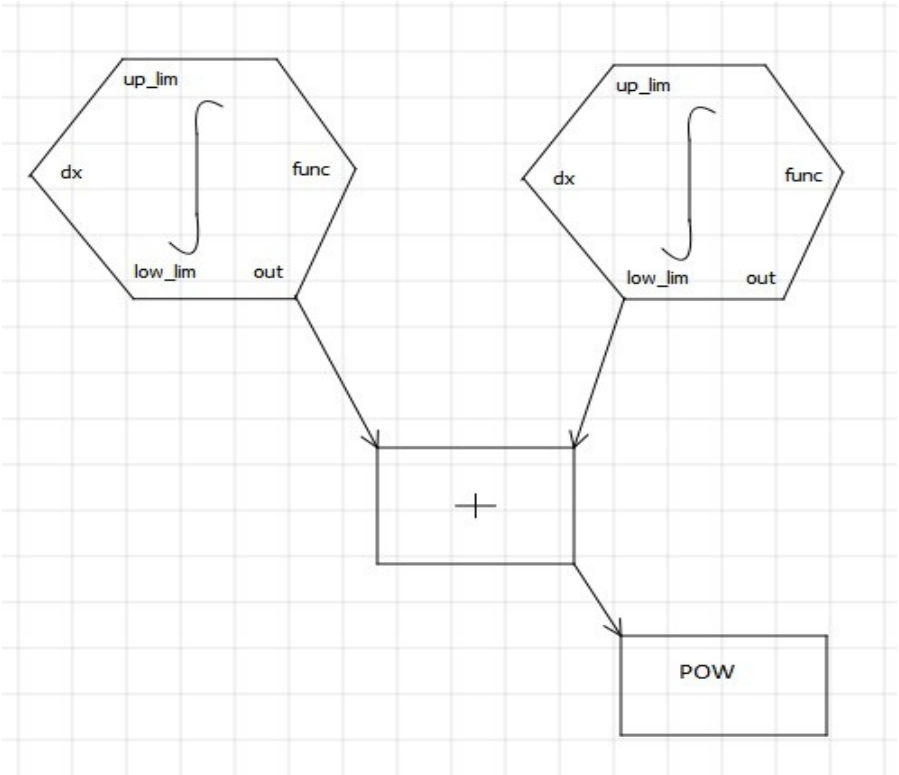

Рис. 4

Алгоритм обхода дерева вычислений будет работать следующим образом:

Будет проведен поиск элементов, не имеющих входящих направленных связей. В нашем случае это два элемента интеграла. Исходящие значения этих блоков будут посчитаны, так как имеются все необходимые значения. Затем из участвующих в поиске блоков и связей будут исключены блоки интеграла и исходящие из них связи. Снова будет произведен поиск элементов, не имеющих входящих направленных связей. Теперь это будет блок суммы. Блок может быть посчитан, так как необходимые значения (два входящих значения интеграла) уже посчитаны. Производится вычисление суммы, и блок суммы с исходящей связью убирается из дерева поиска. Затем проводится очередная итерация поиска элементов, не имеющих входящих направленных связей. Теперь это блок степени, для которого к этому моменту заданы все необходимые свойства. Результат, полученный при вычислении блока степени, будет являться выходным значением программы.

Код на Scilab, соответствующий данной диаграмме будет выглядеть так:

```
x0 = pi/2:pi;

y0 = cos(x0);

res0 = inttrap(x0,y0);          //res0 = -1

x1 = pi:3/2 pi;

y1 = sin(x1);

res1 = inttrap(x1,y1);          //res1 = 1

res2 = res0 + res1;             //res2 = 0

res3 = power(res2, 10);         //res3 = 0
```

Программа на C полученная в результате генерации вернет значение 0.

Заключение

В результате курсовой работы был создан Результатом предметно-ориентированный языка для описания математических расчетов и редактора диаграмм. По диаграммам генерируется код на C, осуществляющий вычисления по заданным правилам. Мы получили для рисования математических диаграмм преимущества CASE-системы: привязку к другим диаграммам, интеграцию с системами контроля версий, удобный специализированный пользовательский интерфейс, возможности интеграции с другими языками и расширяемость за счет метаязыковых средств QReal.

Это позволит в дальнейшем использовать QReal для научных исследований на кафедре теоретической кибернетики, направленных на программное управление роботами. Данная работа позволит им в будущем отказаться от использования дорогостоящих лицензий Matlab и более активно привлекать к своей деятельности учеников физико-математической школы № 239.

У кибернетиков есть достаточно большое количество уже реализованных на Matlab программ и алгоритмов. При переходе на Scilab у них не возникнет необходимости переписывать существующих код, так как Scilab, код на котором генерируется QReal по математическим диаграммам, предоставляет утилиту, позволяющую конвертировать код Matlab в код Scilab.

Дальнейшее направление деятельности будет направлено в первую очередь на увеличение числа доступных диаграмм, поддержку сложных ресурсоемких вычислений и улучшение пользовательского интерфейса. Будет добавлена проверка синтаксической корректности программы. Также будут решаться задачи, связанные с оптимизацией вычислений, так как это актуально для роботов, объем памяти которых сильно ограничен.

Список литературы

1. Терехов А.Н, Брыксин Т.А., Литвинов Ю.В., Смирнов К.К, Никандров Г.А., Иванов В.Ю., Такун Е.И.. Архитектура среды визуального моделирования QReal. //Сборник статей «Системное программирование» 2009, стр 171-196
2. http://doxygen.scilab.org - примеры кода Scilab//20.07.2010
3. http://www.csa.ru/~zebra/my_scilab/sciq_6.html - руководство пользователя Scilab //06.09.2010
4. http://ru.wikipedia.org/wiki/Scilab - статья в википедии о Scilab//15.07.2010
5. http://www.hartes.org/index.php?option=com_content&task=view&id=22&Itemid=40 – описание и исходный код Scilab2c//28.08.2010