Санкт-Петербургский государственный университет
Математико-механический факультет

Среда программирования роботов
QReal:Robots

Курсовая работа студента 245 группы
Тихоновой Марии Валерьевны

Научный руководитель ст.преп. Ю.В.Литвинов

Санкт-Петербург
2012

Оглавление

Введение

QReal:Robots - среда программирования роботов, система для обучения основам программирования и кибернетики. Она базируется на CASE-системе QReal, которая является средством быстрого создания визуальных языков. QReal позволяет создавать предметно-ориентированные визуальные языки, задавать форму используемых фигур и задавать правила генерации кода по диаграмме. Язык программирования роботов является одним из примеров визуального языка, поэтому было принято решение реализовать его с помощью QReal. QReal:Robots - это одно из применений технологии QReal.

Теперь следует пояснить, как системы программирования роботов применяются в образовании. Идея использования реального исполнителя в обучении программированию и кибернетике возникла довольно давно. Обусловлено это тем, что обучение программированию сложно именно из-за того, что объекты, с которыми приходится иметь дело обучающемуся, абстрактны и нематериальны, следовательно, их трудно сравнивать и оценивать. Но нематериальный исполнитель, даже если он отображается на экране, все еще недостаточно нагляден. К тому же, роботы интересны школьникам: процесс сборки робота очень занимателен и увлекает их, им интересно наблюдать, как реальный исполнитель выполняет заданные команды (едет вперед, останавливается, поворачивает). А материал лучше усваивается, когда ученикам интересно; таким образом, понятна польза от применения робототехники в обучении программированию.

Одним из первых, кто оценил проблему сложности работы с нематериальным исполнителем, был Сеймур Пейперт, психолог-информатик из MIT. Он разработал язык программирования LOGO, который позволял видеть результаты работы программы непосредственно - черепашка передвигалась по экрану в соответствии с заданными ей командами. Кроме того, он даже использовал изначально механическую черепашку.

Что касается отечественного подхода к обучению информатике, здесь основоположником считается академик Ершов А.П., вместе с Г.А.Звенигородским и Н.А.Юнерман создавший методики и инструментальные средства (например, Робик и Рапира).

На сегодняшний день уровень развития вычислительной техники позволяет создавать устройства, которые могут как исполнять внутреннюю программу, так и управляться с компьютера по беспроводной связи. Конструкторы на основе контроллеров, поддерживающих эту функциональность, давно внедряются в школьное образование. В частности, одним из наиболее часто используемых конструкторов для обучения робототехнике является Lego Mindstorms.

В стандартном комплекте конструктора поставляется среда программирования роботов NXT-G, однако у нее имеется ряд недостатков, главным из которых является ее недостаточная функциональность (не поддерживается сложная математика). Поэтому преподаватели отдают предпочтение среде программирования Robolab, которая позволяет создавать гораздо более сложные программы, содержащие нетривиальные математические выражения. Но в Robolab недостаточно полный и качественный перевод на русский язык, он долгое время не поддерживается и обладает устаревшим и недостаточно дружественным пользовательским интерфейсом. К тому же, Robolab является платной средой программирования.

QReal:Robots является свободным средством программирования роботов. В нем есть такие особенности, как пошаговая отладка программы и переход от диаграммы поведения робота к текстовому представлению, что актуально для тех, кто больше интересуется программированием, нежели кибернетикой. На момент начала нашей работы над этим продуктом большая часть функциональности уже была реализована, но была недоработана и содержала большое количество ошибок. Наша задача состояла в том, чтобы исправить эти ошибки и довести продукт до состояния, в котором его можно было бы уже внедрять в школах. Для этого мы ездили на робототехнический кружок в Лицей №239 и общались со школьниками, выясняя, что их не устраивает в QReal:Robots; также мы принимали участие в городских соревнованиях по робототехнике.

Работа над усовершенствованием среды программирования QReal:Robots поделилась на две части: устранение недостатков, связанных с взаимодействием с роботом, и исправление ошибок в генерации кода; последнее и стало темой этой курсовой работы.

Обзор существующих сред программирования

На сегодняшний день наиболее часто используемыми системами для программирования роботов являются Robolab, NXT-G и RobotC.

NXT-G - графическая среда программирования роботов. Создание программы для управления поведением робота в ней похоже на рисование блок-схемы: с помощью имеющихся блоков рисуется диаграмма и задаются характеристики поведения робота. NXT-G очень проста для освоения и использования из-за своей наглядности, но ее существенным недостатком является ее недостаточная функциональность.

Robolab - тоже графическая среда программирования, самая распространенная в школах и вузах для преподавания кибернетики. Она поддерживает возможность создавать гораздо более сложные программы, но имеет ряд недостатков, о которых уже было сказано выше.

RobotC - текстовая среда программирования, позволяющая создавать программы для управления поведением робота с помощью языка программирования С. У нее есть два режима работы: базовый режим и режим для специалистов (в базовом режиме некоторая функциональность отсутствует). RobotC имеет мощный интерактивный отладчик в режиме реального времени; с помощью этой среды можно создавать сложные и эффективные программы, но текстовое программирование может оказаться недостаточно наглядным.

QReal:Robots

QReal:Robots разрабатывается силами сотрудников и студентов кафедры системного программирования. Система постоянно развивается с учетом замечаний и пожеланий преподавателей школ и вузов. На момент начала нашей работы над средой QReal:Robots в ней уже была реализована основная функциональность для управления роботом. Система позволяла создавать графические программы в виде последовательности блоков, соединенных линиями управления, и исполнять эти программы на компьютере, посылая роботу команды через Bluetooth-интерфейс. Были реализованы блоки, отвечающие за аппаратную часть - управление датчиками цвета и нажатия, ультразвуковым датчиком расстояния, управление моторами и динамиками, а

также блоки, задающие логику работы программы - условные операторы, проверяющие показания датчиков или значения переменных, циклы и средства для работы с таймерами. Была осуществлена поддержка сложных математических выражений, состоящих из переменных, констант, арифметических операций, тригонометрических функций и значений датчиков.

Кроме того, важной особенностью QReal:Robots является наличие двухмерной модели робота, способной исполнять те же программы, что и реальный робот. Она может помочь при отладке, позволяя отлаживать программу без доступа к реальному роботу. Двухмерная модель в QReal:Robots, по сути, повторяет функциональность черепашки LOGO, но, кроме этого, она поддерживает сенсоры, и в этом заключается ее существенное преимущество.

Возникшие задачи

Тем не менее, чтобы начать внедрять QReal:Robots, требовалось решить целый ряд задач, о которых и пойдет речь далее. Задачи, в основном, были связаны с недостаточностью поддержки некоторых аппаратных компонентов робототехнического конструктора --- поскольку на первых этапах разработки задача поддержки всех видов датчиков для NXT во всех режимах и не ставилась. Робот должен был устойчиво функционировать с теми датчиками, что входили в стандартную поставку конструктора NXT 2.0, однако же, для решения реальных задач этого оказалось мало. Кроме того, при столкновении с реальными задачами выявились и другие слабые места продукта --- некоторые неточности и концептуальные ошибки, допускаемые генератором кода для ОС nxtOSEK, связанные, опять же, с необходимостью быстро представить работающее на некоторых частных случаях решение. Таким образом, наша задача состояла в том, чтобы поддержать необходимые для решения реальных задач датчики. Например, для реализации алгоритма движения по линии нам понадобились датчики цвета; но оказалось, что nxtOSEK не поддерживает датчики цвета, и пришлось использовать вместо них датчики света. Эти датчики не умеют распознавать цвета, а распознают только интенсивность света, но поддерживаются nxtOSEK; их поддержку в QReal:Robots тоже нужно было реализовать. Также нужно было исправить ошибки в генерации кода; в частности, нужно было поддержать генерацию функций-инициализаторов (функций, в которых инициализируются сенсоры), генерацию переменных в начале программы и

сделать так, чтобы код генерировался в соответствии с инициализированными сенсорами. К тому же, выяснилось, что заливка программы на робота по USB для выполнения непосредственно на роботе не всегда работает корректно, потребовалось поправить и это.

Цели

Целью данной курсовой работы стало исправление ошибок в генерации кода, выявленных при столкновении с реальными задачами, и добавление новой функциональности, необходимой для участия в соревнованиях.

Процесс генерации кода

Программа в QReal:Robots представляется в виде последовательности блоков, соединенных связями. По этой последовательности генерируется код, который заливается на робота. На роботе должна стоять операционная система nxtOSEK, являющаяся самой быстрой на данный момент ОСРВ для NXT и к тому же свободно распространяемой. Эта система поставляется вместе с набором инструментов, которые позволяют скомпилировать файл на C в бинарный файл, исполняемый на роботе, и залить его на робот.

Файлы на C генерируются следующим образом: в QReal:Robots есть генератор, который получает на вход диаграмму, хранящуюся в репозитории, разбирает ее и генерирует по каждому блоку соответствующий код. Он обладает возможностью не только генерировать код по элементарным блокам, но и разбирать логические конструкции, например, циклы и условные операторы.

Реализация

В ходе работы было реализовано следующее:

1. Генерация функций инициализации.

В nxtOSEK есть возможность запускать некоторые функции до запуска основной программы. Эти функции предназначены прежде всего для инициализации сенсоров

робота. Кроме того, выяснилось, что без этих функций некоторые сенсоры не работали и показывали неправильные значения. Чтобы реализовать генерацию этих функций, в шаблон была добавлена строчка, которая при генерации заменяется на их реализацию. Тело функции инициализации генерируется следующим образом: из блока инициализации берутся сенсоры, которые нужно инициализировать, генерируется соответствующий код. Из блока функции, у которого стоит галочка “функция инициализации”, берется произвольный код, который тоже можно добавить в функцию инициализации.

2. Генерация переменных в начале программы.

Выяснилось, что некоторые переменные требуется генерировать в самом начале, вне основной функции, чтобы они были видны не только ей, а, например, в функциях инициализации. В шаблон добавлена строчка, которая при генерации заменяется на переменные, генерирующиеся при разборе диаграммы по блокам функции.

3. Корректная генерация для сенсоров.

Были выявлены ошибки в генерации для сенсоров, например, оказалось, что генерация не учитывает выбранные в блоке инициализации сенсоры и генерирует функции всегда для сенсора касания. В зависимости от выбранных в этом блоке сенсоров нужно было заменять переменные сенсоров на соответствующие им функции.

4. Новые блоки для сегвея, их поддержка в генераторе.

Для того, чтобы сделать представление проекта на соревнованиях более интересным, было принято решение продемонстрировать реализацию алгоритма балансировки сегвея. Сама задача балансировки сегвея является хоть и стандартной в робототехнике, но тем не менее труднореализуемой. Трудность заключается в том, что, во-первых, алгоритм достаточно сложен, а во-вторых, программу нельзя отлаживать из-за того, что данные по Bluetooth и по USB передаются с задержкой, поэтому робот не успевает реагировать на изменение наклона и не балансирует. Нужно было реализовать алгоритм в QReal:Robots, для чего потребовалось поддержать новые блоки (в частности, блоки инициализации и балансировки). Получившаяся в результате диаграмма поведения робота достаточно проста, таким образом, было показано, что даже столь сложные задачи, как балансировка сегвея, с помощью QReal:Robots могут быть довольно просто реализованы.

Для реализации задачи балансировки сегвея была использована стандартная библиотека. Было принято решение распространять файлы для сегвея вместе с дистрибутивом QReal, чтобы при генерации кода генератор перекладывал нужные файлы из ресурсов. Однако же, всегда генерировать код, связанный с сегвеем, даже когда в программе не используются соответствующие блоки, не нужно; таким образом, было решено просматривать диаграмму на предмет наличия в ней блоков балансировки и инициализации балансировки и в зависимости от этого генерировать или не генерировать нужный код.

5. Исправление ошибок.

Были исправлены некоторые ошибки, не связанные с генерацией, например, аварийное завершение программы при удалении и повторном добавлении сенсоров в 2д-модели.

Заключение

В процессе работы над поставленными задачами было сделано следующее:

- сделан обзор существующих сред программирования
- исправлен ряд ошибок, связанных с генерацией кода
- добавлены блоки, потребовавшиеся для реализации алгоритма балансировки сегвея
- проведено тестирование системы на реальных задачах
- система продемонстрирована на городских соревнованиях Санкт-Петербурга по робототехнике и на робототехническом фестивале в Москве

Список литературы

1. Брыксин Т.А., Литвинов Ю.В., Среда визуального программирования роботов QReal:Robots // Материалы международной конференции "Информационные технологии в образовании и науке". Самара. 2011. С. 332-334.
2. С. А. Филиппов, Робототехника для детей и родителей//СПб, Наука, 2010 г.
3. QReal: Robots, URL: http://se.math.spbu.ru/SE/qreal.