

1 Титульник

2 Модели исполнения

Одной из общепринятых и общеизвестных моделей исполнения является модель потока управления. Если мы говорим о визуальных языках, то здесь мы традиционно используем блоки исполнения и для их активации передаем токен управления. Альтернативой является модель потока данных. Здесь для активации блоков используются данные, которые генерируются и обрабатываются блоками. Языки программирования основанные на такой модели называются реактивными или потоковыми, они же *dataflow languages*.

3 Управление роботом

Систему управления роботом можно рассмотреть как последовательное взаимодействие 3 составляющих: сенсоры генерируют данные, логика системы управления преобразует эту информацию в импульсы и передает на приводы. Это по сути задача трансформации данных, т.е. система управления роботом может рассматриваться как класс реактивных систем.

4 Модели декомпозиции управления роботом

До выхода в 1986 году статьи Родни Брукса декомпозировать систему вправления роботов было принято горизонтально, Брукс предложил декомпозировать систему на основе поведений, когда каждое поведение использует значения сенсоров и управляет приводами робота независимо от других. Это позволяет легче тестировать и быстрее создавать системы управления (например разные группы могут независимо создавать уровни поведения).

5 Архитектура системы управления роботом

Так же Брукс предложил архитектуру для своей системы управления, где верхние уровни поведения имеют больший приоритет за счет возможности подмены или остановки данных в потоке, которые генерируют нижние уровни. Её также называют архитектура Категорий. Эта архитектура является одной из самых популярных для построения систем управления роботом.

6 Исследования в области

Реактивные системы управления роботом хорошо выражается потоковыми языками. Об этом много говорят и пишу в науке. В первой работе исследовательская группа из Германии описывает инструмент на основе сложных формализмов для создания и анализа сложных систем управления роботом на основе событийно-ориентированного подхода. Нам хочется создать что-то похожее, но более простое в использовании.

Во второй группа из Англии описывает прототип инструмента визуального программирования одного конкретного робота на основе архитектуры Брукса и говорит об удобствах использования визуальных языков для программирования роботов, используя эту архитектуру.

На конференции IEEE-ишной конференции по визуальным языкам VL/HCC была представлена интересная работа Новозеландской группы исследователей, чья идея наиболее близка к нашей, но результаты в открытом доступе отсутствуют.

Не говори: у них просто идея, у меня идея и инструмент.

7 Среда программирования роботов

это же доказывается количеством инструментов для программирования в терминах потоков данных, активно использующихся в индустрии. Учебные среды для программирования используют более простую модель потока управления. Нам интересно пересечение этих инструментов. Дополнительная «ступень» в виде простого в освоении потокового языка была бы весьма полезной при переходе от учебных языков программирования к тем, что используются в индустрии.

8 Постановка задачи

Целью данной квалификационной работы является расширение TRIK студии новым языком программирования роботов. С возможностью интерпретации программ написанных на нем как в режиме двумерной модели, так и на реальном роботе. А также апробация нового языка на типовых задачах управления роботом.

НЕ НАДО Среда находится в открытом доступе и обладает легко масштабируемой архитектурой. Которая позволяет расширить себя новым визуальным языком и переиспользовать кодовую базу таких «рутинных» операций, как взаимодействие с роботом. Что позволяет сосредоточиться на поставленных задачах, а не на задачах коммуникации с роботом.

9 Язык

Два основных элемента языка это связь (реализующая поток для данных передаваемых между блоками) и блок. На блоках слева и снизу могут быть входные порты для данных. Справа выходные порты. Сверху могут быть отдельные порты для явной передачи контроля управления. Также в блоке может быть одно или несколько текстовых полей, например условий или кода на языке Lua. Порты блоков подписываются. Блоки можно поделить на несколько групп. За неимением времени рассмотрим только некоторых представителей некоторых групп, необходимых для примеров далее.

4 входных порта моторов передают импульсы на соответствующие моторы робота, сенсор постоянно генерирует обновленные данные, геймпад генерирует данные соответствующие действиям пользователя

Блоки запаковки и распаковки позволяют собирать разные данные в 1 набор или разбивать на составляющие. Блок подавления позволяет замещать данные в течение предустановленного времени, таймер запускается первым набором от подавляющего блока и не может быть обновлен пока не сбросится до 0.

Блок распараллеливания явно распараллеливает программу. Пользовательский блок нужен для вызова включения в программу другой программы в качестве нового блока, в включаемой программе используется выходной порт ассоциированный с портом Пользовательского блока.

Мы можем устанавливать глобальные переменные, изменять их через входной порт, отправлять ее значение как данные. Чтобы закончить работу системы управления в языке присутствует завершающий исполнение блок. Для фильтрации данных по времени и или количеству пропускаемых данных и или условию есть блок Фильтр. Для выполнения математических операций или участков кода в языке присутствует блок Текстового программирования. Всего в языке около 50 различных блоков, их детальное обсуждение заняло бы слишком много времени.

10 Реализация

Для реализации задачи в систему было добавлено 2 плагина: один описывает визуальный язык и предоставляет визуальный редактор языка для системы, он содержит метамодель языка и полностью сгенерирован с помощью модельно-ориентированного подхода на базе DSM платформы QReal, второй содержит реализацию интерпретатора диаграмм роботов, созданных в реактивном стиле. Он принимает диаграмму, созданную первым плагином и преобразует ее в последовательность команд для целевого робота. Созданные мною компоненты выделены зеленым цветом.

11 Апробация: ПД-регулятор для движения вдоль стены

11.1 Система управления

Рассмотрим примеры созданные на новом языке: к примеру, в отличие от потокового языка, можно создать пропорционально-дифференциальный регулятор для езды вдоль стены с помощью всего 4 блоков. На этом слайде представлена интерпретация данной программы на двумерной имитационной модели.

11.2 Интерпретация на модели

На этом слайде представлена интерпретация данной программы на двумерной имитационной модели.

11.3 Интерпретация на роботе

На этом слайде представлена интерпретация данной программы на роботе.

12 Апробация: двухуровневая система управления роботом

12.1 операторский контроль

Давайте рассмотрим более сложную систему управления роботом. Допустим, мы хотим управлять роботом с пульта, в данном случае с геймпада, и для простоты сделаем допущение, что нажатие любой кнопки кроме манипулирования рычагом направления выключит робота. Координаты рычага направления преобразуются блоком текстового программирования в импульсы для моторов и передаются на выходной порт, он нужен, чтобы использовать эту программу как новый блок, который выдает импульсы для моторов.

12.2 Избегание столкновений

Вспомним, что робот это дорогая игрушка, поэтому добавим еще один аспект поведения робота: пусть робот не позволит нам столкнуться с препятствием при управлении с пульта. Обе задачи будут запущены параллельно. Здесь мы собираем значения с двух сенсоров в 1 массив и передаем его фильтру, который проверит условие и, если робот будет близок к столкновению, передаст эти значения дальше блоку текстового программирования, который сгенерирует на основе значений с сенсоров импульсы для моторов.

12.3 Управление на основе архитектуры категорий

Теперь у нас есть 2 поведения, причем мы хотим, чтобы при возникновении опасности один из них подавлял второй. Запустим оба поведения параллельно, и добавим блок подавления и замещения, который при приходе данных от подавляющего, не пропускает данные подавляемого в течение предустановленного времени, затем мы распаковываем значения импульсов для каждого мотора и подаем их на моторы.

13 Результаты

Еще раз повторю результаты работы: создан новый потоковый визуальный язык, а также редактор и интерпретатор для него, чья работа была проведена на типовых задачах управления роботом. Также по результатам работы написаны две публикации на конференции.