

описания предметной области выделяют информационные сущности, их атрибуты и связи между сущностями. В качестве атрибутов используются исключительно информационные атрибуты, то есть те которые несут информационную нагрузку о свойствах сущности.

Для каждого атрибута сущностей необходимо определить их домены. Домен – это выражение, определяющее значения, разрешенные для данного атрибута. На основании информации о домене атрибута на этапе логического проектирования будут выбираться типы полей для хранения данных атрибута экземпляра сущности.

Для наглядного представления полученной модели предметной области используется диаграмма сущность-связь. На данный момент предложено множество вариантов подобных диаграмм, но на этапе концептуального проектирования наиболее удобной можно считать диаграмму, предложенную одним из основоположников реляционной модели базы данных Питером Ченом.

На основании построенной диаграммы делается выбор модели базы данных.

Второй этап проектирования базы данных – логический. На этом этапе осуществляется преобразование концептуальной модели в виде нотации Чена в диаграмму классов UML.

Заключительным является этап физического проектирования. На нем проектирование осуществляется по правилам конкретной СУБД и её диалектом языка SQL. Задачи физического этапа проектирования можно кратко сформулировать так: выбор типов данных и правильное их применение.

Описание процесса трехэтапного проектирования баз данных представленное выше несколько упрощенное. Однако оно показывает как разработать базу данных практически с нуля, при этом явно не используя правила нормализации, а лишь проверяя полученные схемы на соответствие нормальным формам, что заметно упрощает процесс проектирования.

УДК 004.42+004.896

ПРОГРАММИРОВАНИЕ МОБИЛЬНЫХ РОБОТОВ LEGO MINDSTORMS EV3

Добыш Д.С., студ., Черненко Д.В., ст. преп.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Основная задача данной работы – написание программы контроллера робота Lego Mindstorms EV3 для его прохождения по лабиринту, выложенному черной изолентой. Программирование велось на языке C для специальной ОС ev3dev, основанной на Debian Linux.

Построенный робот представляет собой базовую модель ev3 mindstorms robot, но использующую USB-порт для подключения двух периферийных устройств – веб-камеры и WiFi-адаптера.

Поставленная задача была разделена на три этапа: получение правильной картинки лабиринта, вычисление маршрута и его выполнение.

Получение картинки лабиринта выполнялось с использованием API video4linux. После получения картинки возникла проблема искажения перспективы. Камера, установленная на роботе, не имеет функции изменения фокуса. Поэтому проект был разделен на две отдельные программы: одна для ПК, предоставляющая большую производительность

и возможность пользовательского ввода; вторая – непосредственно для контроллера. Программы общаются между собой посредством сети WiFi. Исправление искажения в первой программе осуществляется на OpenGL с использованием 3D-пространства и ручной настройки.

Для *высчитывания маршрута* написан алгоритм поиска пути, схожий с алгоритмом D*. Для упрощения и сокращения времени работы, высчитывание маршрута также производится на ПК. Полученный маршрут разделяется на участки, а потом на список из двух типов команд: повернуть и ехать.

Последним этапом, контроллер получает список команд и сохраняет их в памяти для последовательного выполнения. Включение двигателя осуществляется записью определённых данных в файлы устройств, для чего была использована библиотека языка C "ev3dev-c". В этой библиотеке представлены различные методы для управления двигателями, каждый из которых существует для установки определённых настроек (полярность, количество поворотов и др.).

Перед запуском двигателя необходимо установить желаемые величины и режим работы. В каждом двигателе есть датчик угла поворота, по которому можно определить прогресс выполнения команды, что позволяет выполнять их одну за другой по списку.