



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАЗРАБОТКА УЧЕБНОЙ ПРОГРАММНОЙ ПЛАТФОРМЫ ДЛЯ РОБОТА iROBOT CREATE

Бармина Е.И.* , Ланин В.В.* , Плетнёв А.О.**

* *Пермский филиал национального исследовательского университета
«Высшая школа экономики», г. Пермь, Россия*

enokokok@ya.ru

vlanin@live.com

** *Пермский государственный национальный исследовательский университет, г. Пермь, Россия*

artem.o.pletnev@gmail.com

Статья посвящена разработке учебного программного комплекса для исследования алгоритмов движения мобильного робота. В соответствии с предлагаемым подходом, алгоритмы основываются на концептуальном описании пространства, в котором передвигается робот. Прототип программный комплекс разрабатывается для мобильного робота iRobot Create на платформе Java.

Ключевые слова: мобильный робот; алгоритм движения; онтология пространства.

Введение

В рамках проекта планируется разработка программного комплекса для разработки алгоритмов движения мобильного робота на основе формальной концептуальной модели окружающего мира. Используемые на данный момент подходы к построению алгоритмов в большинстве случаев не допускают расширения и динамической настройки. В предлагаемом подходе алгоритм передвижения не закодирован жестко, он строится на базе вывода по базе знаний. Наличие такой интеллектуальной системы ориентации, позволит роботу планировать свои действия по перемещению в пространстве, просчитывать оптимальный маршрут движения, собирать, обобщать и использовать информацию об окружающем его пространстве и встречающихся препятствиях.

На данный момент существует несколько подходов к представлению окружающей местности на карте, используемой при движении робота: растровая карта, таблица связей, объектная карта и др. Растровая карта хранится как двумерный массив, показывающий доступные и закрытые для прохождения роботом зоны. При использовании таблицы связей предполагается, что в основу заложена карта связей, хранящая некие узлы на карте и связи между ними. Объектная карта – это попытка эмуляции способа мышления человека и

его подход к ориентации в пространстве [Bouten, 2012]. Данный подход предполагает, что робот, определив несколько ключевых точек карты, получает возможность самостоятельно прокладывать путь к нужной точке. Также необходимо уточнить, что алгоритмы зависят от аппаратных средств робота.

1. Платформа iRobot Create и поддерживающие ее инструментальные средства

1.1. Мобильный робот iRobot Create

Для реализации алгоритма отлично подойдет робот, выпущенный CES iRobot в 2007 году - iRobot Create, основанный на платформе Roomba. Он создан специально для разработчиков роботов и дает возможность собирать собственных роботов, писать для них программы, на существующей платформе разрабатывать свои собственные модели роботов [Isaacs, 2011]. Данный робот включает 4 контактных сенсора стены и всенаправленный инфракрасный приемник, расположенный в передней част робота.

Открытый интерфейс робота состоит из электронного и программного интерфейсов [iRobot, 2006]. Данные интерфейсы контролируют поведение и считывают информацию с сенсоров. Программный интерфейс позволяет манипулировать

поведением конструктора. Он считывает информацию с сенсоров при помощи серии команд, включает команды режима, команды привода, звуковые команды, демонстрационные команды и команды опроса сенсоров. Эти команды посылаются на последовательный порт конструктора с компьютера или микроконтроллера посредством Mini-DIN разъема или DB-25 разъема грузовой отсека».



Рисунок 1 – Устройство робота

Следует заметить, что iRobot Create оснащен минимальным набором аппаратных средств, и чего следует, что алгоритмы разработанные для данной модели могут быть перенесены и на более совершенные устройства.

1.2. Инструментальные средства разработки для роботов

На данный момент наиболее популярными программными решениями для разработки, управления и программирования роботов являются ROS, MRDS, Player.

1.2.1. Система ROS

ROS (Robot Operating System) – операционная система с открытым исходным кодом для роботов (лицензия BSD). Проект разрабатывается исследовательской лабораторией Willow Garage в сотрудничестве с университетом Стэнфорда. ROS реализует системный уровень управления роботом, а на его основе развиваются прикладные пакеты: библиотека машинного зрения OpenCV, система планирования действий, сервер управления Player и другие технологии, используемые в десятках научных и прикладных проектов. С точки зрения архитектуры ROS – это распределённая система процессов (узлов), которые могут быть сгруппированы в так называемые пакеты и стеки, пригодные для распространения. ROS легко интегрируется с другими программными компонентами (на данный момент с OpenRAVE, OROCOS и Player). ROS не зависит от языка программирования, уже реализованы версии на C++, Python, LISP, Octave Java, Lua. ROS имеет встроенный пакет для тестирования приложений. При создании программ на ROS строится «граф» – сеть

точка-точка (peer-to-peer network) из процессов, которые связываются друг с другом через инфраструктуру ROS.

1.2.2. Microsoft Robotics Developer Studio

Microsoft Robotics Developer Studio (Microsoft RDS, MRDS) – Windows-ориентированная среда разработки приложений для роботизированных платформ [Гай, 2012]. В состав Robotics Studio входят следующие компоненты: runtime environment, VPL, simulation environment. *Runtime environment* – окружение, в котором выполняется приложение для роботов, происходит отслеживание и взаимодействие с другими приложениями для роботов. В основе Runtime environment лежит CLR 2.0, что дает возможность разрабатывать приложения на языках программирования платформы Microsoft .NET. Runtime environment состоит из двух элементов: CCR (concurrency and coordination runtime) – библиотека для работы с параллельными и асинхронными потоками данных и DSS (Decentralized Software Services) – средство создания распределенных приложений на основе сервисов, базирующиеся на протоколе Decentralized System Services Protocol (DSSP). *VPL* (visual programming language) – визуальный предметно-ориентированный язык программирования для разработки приложений для роботов. *Simulation environment* – окружение для выполнения приложения для роботов в симулируемых условиях. Приложение в Robotics Studio – это композиция слабосвязанных параллельно выполняющихся компонентов. При этом все компоненты в Robotics Studio – это независимо исполняемые сервисы, т.е., например, для разработчика программы не существует физического мотора, а есть сервис с интерфейсом, к которому нужно обратиться, чтобы работать с мотором из написанной программы.

К существенным недостаткам RDS можно отнести зависимость от Windows и закрытый исходный код. Кроме того, Robotics Developer Studio не имеет встроенных систем компьютерного зрения, навигации и машинного обучения. И протокол SOAP, используемый для взаимодействия распределенных сервисов, плохо подходит для приложений, работающих в режиме реального времени.

1.2.3. Проект Player

Проект Player (ранее проект Player/Stage или проект Player/Stage/Gazebo) представляет собой инструментальный по созданию свободного программного обеспечения для исследования робототехнических и сенсорных систем. Проект состоит из 3 основных компонентов: сервера Player и платформ для симуляции роботов, двумерного симулятора Stage и трёхмерного симулятора Gazebo. Можно сказать, что Player наиболее часто используемый интерфейс в робототехнических исследованиях. Большинство ведущих журналов о робототехнике, регулярно публикуют статьи о применении Player (Stage и Gazebo) для управления

и моделирования робототехнических экспериментов (в академических, правительственных и промышленных лабораториях). Player обеспечивает сетевой интерфейс для различных роботов и сенсорного оборудования. Клиент-серверная модель, реализованная в Player, позволяет программам управления робота быть написанными на любом языке программирования и работать на любом компьютере, подключенного к сети вместе с роботом.

Компоненты проекта работают на POSIX-совместимых операционных системах, включая Linux, Mac OS X, Solaris и BSD; планируется портирование на Microsoft Windows. Проект был основан в 2000 году в Университете Южной Калифорнии в Лос-Анджелесе и широко используется в робототехнических исследованиях и обучении. Программное обеспечение распространяется под лицензией GNU General Public License с документацией под GNU Free Documentation License.

2. Алгоритм движения робота на основе онтологического подхода

В предлагаемом подходе карта для движения робота индексируется понятиями онтологии, в которой описаны возможные препятствия и способы их преодоления. Кроме возможных препятствий, в онтологии описываются такие объекты как само помещение, дверной проем, ступени, пороги, тип напольного покрытия и пр. Таким образом, для решения конкретной задачи перемещения, робот сначала строит или загружает построенную ранее карту, а затем прокладывает на ней маршрут, интерпретируя понятия представленных на карте объектов.

2.1. Онтология

Для того, чтобы робот мог ориентироваться в пространстве, ему нужно ввести некоторые понятия, такие как - что такое комната, какие объекты она может содержать, какие у неё есть свойства и т.д. Для описания всех этих параметров необходима онтология. Онтология – формальное явное описание понятий в рассматриваемой предметной области, свойств каждого понятия, описывающих различные свойства и атрибуты понятия, и ограничений, наложенных на свойства. Онтология вместе с набором индивидуальных экземпляров классов образует базу знаний.

Начнем построение онтологии с описания классов. Для робота, помещение рассматривается как некоторая среда, наполненная объектами, которые нужно распознать. Следовательно, онтология должна описывать отличительные черты этих объектов с точки зрения робота (форма объектов, их размер, материал и пр.). Начнем с того, что помещение – это не только расположенные в нем объекты, но и его геометрическая форма, так же оно имеет выходы. Объекты, расположенные в

помещении, прежде всего отличаются по геометрической форме и по размерам. Исходя из этих параметров можно сделать предположение об объекте. Так же необходимо описать отношения между классами (например, дизъюнктивные классы и т.д.). Далее осталось описать свойства объектов. Объекты в помещении (мебель и различные препятствия) тоже имеют свои параметры, такие как: периметр, координаты внутри помещения, диаметр, и др. Заключительный шаг в создании онтологии - создание конкретных экземпляров классов. Например: если объект имеет четыре опоры, а его периметр около 200 см, то, скорее всего, это стул. На данный момент прототип онтологии описан на языке OWL 2.0 с использованием системы Protégé (рис. 2).

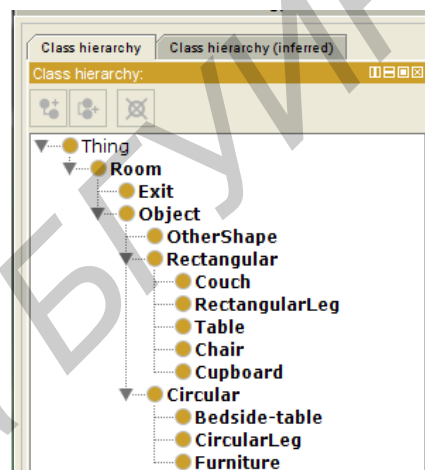


Рисунок 2 – Фрагмент иерархии классов онтологии

2.2. Алгоритм движения робота

Основные компоненты программно-технической системы представлены на рис. 3. Ключевым элементом системы является управляющий модуль, он обрабатывает информацию, полученную от сенсоров робота, и передает управляющие сигналы. Обрабатывая информацию, управляющий модуль делает запросы к онтологии для идентификации объектов, если же объект идентифицирован ранее, то информация извлекается из карты.

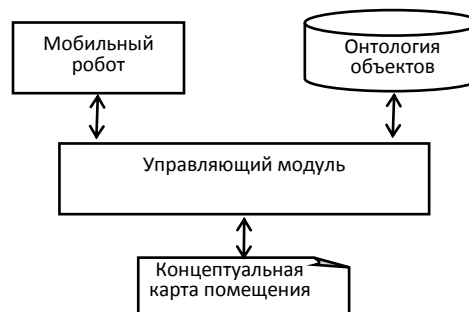


Рисунок 3 – Фрагмент иерархии классов онтологии

Перемещение робота может происходить в нескольких режимах: построение концептуальной карты помещения, поиск роботом целевого объекта и комбинированный режим (поиск с построением карты, см рис. 4).

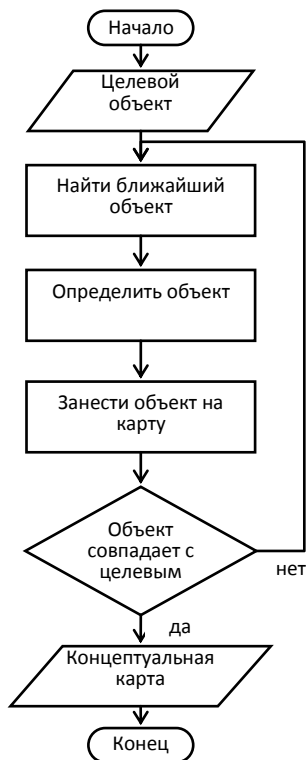


Рисунок 4 – Фрагмент иерархии классов онтологии

Заключение

Особенностью предлагаемого подхода является возможность расширения и динамической настройки алгоритма, за счет расширения онтологической базы знаний объектов пространства. На данный момент разработан прототип онтологии объектов и библиотека коммуникации с iRobot Create по протоколу Bluetooth. Платформой реализации является Java, что позволяет получить кроссплатформенное решение. В дальнейшем при развитии проекта планируется использовать систему ROS. Разработанный программный комплекс планируется использовать в учебном процессе в курсах, связанных с изучением интеллектуальных систем и онтологического инжиниринга.

Библиографический список

- [iRobot, 2006] iRobot Create Open Interface (OI) Specification, http://www.robotmart.ru/product_images/creator_Series_Manual.pdf
- [Гай, 2012] Гай В.Е. Microsoft Robotics Developer Studio. Программирование алгоритмов управления роботами.– М.: ЭКОМ Паблицерз, 2012. – 184 с.
- [Bouten, 2012] Ontology-Driven dynamic discovery and distributed coordination of a robot swarm.– Bouten N., Hristoskova A., Ongenaе F., Nelis J., Turck F.//Dependable Networks and Services Lecture Notes in Computer Science Volume 7279, 2012, pp 2-13.
- [Isaacs, 2011] A Guided Internship For High School Students Using iRobot Create Isaacs J., Klein D., Hespanha J.– 18th IFAC World Congress.– 2011.– V.16, pp. 12820- 12825.

EDUCATIONAL SOFTWARE PLATFORM DEVELOPMENT FOR IROBOT CREATE

Barmina E. *, Lanin V. *, Pletnev A. **

* Perm Branch of the National Research University Higher School of Economics, Perm, Russia

enokokok@ya.ru, vlanin@live.com

** Perm State National Research University, Perm, Russia

artem.o.pletnev@gmail.com

Article is about the development of an educational software for a mobile robot motion algorithms research. According to the proposed approach, the algorithms are based on the specially developed ontological resources that describe the robotic movement space. Software package is developed for the mobile robot iRobot Create.

Within the project the development of a software package for the development of a mobile robot motion algorithms, which are based on a formal conceptual model of the world planned. An already developed algorithms construction approaches can not be extended and configured dynamically. The proposed approach assumes that the movement algorithm is not strictly coded and is built on the basis of the ontological knowledge base output. Such intellectual orientation system will allow the robot to plan the steps to move in the space, to calculate the optimal route and to collect and use information about the surrounding obstacles.

At the moment, there are many different approaches to display the surrounding area on the map: raster map, table relationships, the object map and others. Raster map is stored as a two-dimensional array, which showing the areas, which are closed or available for the robot passage. If the linked table is used, it is assumed that the map of connections, which stores some nodes on the map and the links between them, is underlie. The object map is an attempt to emulate the human way of thinking and orientation in space, which assumes that the robot, having learned a few key points on the map, gets an opportunity to pave the way to the desired point. It is assumed that the algorithm is developed for a robot equipped with a minimum set of a hardware, that is the iRobot Create in the base set.

The proposed approach implies that the robot motion map is indexed by concepts of ontology which is developed on the OWL language and describes the possible obstacles and ways to overcome it. In addition, it describes such objects as the room itself, the doorway, the steps, the thresholds, the type of flooring, and so on. Thus, for a particular movement purpose, firstly the robot builds or loads the map, which is constructed before, and then, interpreting the concept of the map objects, paves the route.

Now, the prototype of an objects ontology and a library communication have been completed. The platform of implementation is Java. The developed software system will be used in the learning process in the courses related to the study of intelligent systems and ontological engineering.