

учета. Двойная запись, по сути, показывает внутреннюю связь объектов, экономический смысл и содержание каждой операции, переход средств из одного состояния в другое. В бухгалтерском учете взаимосвязь счетов в проводке называют корреспонденцией, а счета, составляющие проводку, корреспондирующийся.

Таким образом, бухгалтерская проводка представляет собой корреспонденцию счетов, когда одновременно делается запись по дебету и кредиту счетов на сумму хозяйственной операции, подлежащей регистрации.

Литература

1. Радченко, М.Г. 1С:Предприятие 8.2. Практическое пособие разработчика / М.Г. Радченко, Е.Ю. Хрусталёва. – М.: ООО «1С – Паблишинг», 2009. – 872 с.

2. Тимофеев, Г.С. Конфигурирование и администрирование 1С:Предприятие / Г.С. Тимофеев, Д.С. Шумейко. – М.: Феникс, 2003. – 320 с.

3. Когаловский, М.Р. Энциклопедия технологий баз данных / М.Р. Когаловский. – М.: Финансы и статистика, 2003. – 800 с.

4. Рязанцева, Н.А. 1С:Предприятие. Бухгалтерский учет. Секреты работы / Н.А. Рязанцева, Д.Н. Рязанцев. – СПб.: БХВ – Петербург, 2004. – 350 с.

Д.Е. Храбров (УО «ГГТУ им. П.О. Сухого», Гомель)

Науч. рук. **И.А. Мурашко**, д-р техн. наук, доцент

МЕТОДИКА ЛОКАЛЬНОГО WI-FI ПОЗИЦИОНИРОВАНИЯ НА ОСНОВАНИИ АЛГОРИТМА СОПОСТАВЛЕНИЯ С ОБРАЗЦОМ

GPS навигация не подходит для многоэтажных зданий, а значит необходимо использовать другую технологию. Предлагается методика позиционирования объекта внутри организаций на основе *Wi-Fi*.

Основные положения предлагаемой методики локального позиционирования можно разбить на два этапа [1]. Предварительный этап – этап обучения системы, ведь для расчёта необходимо иметь эталонный набор точек с уже известными координатами. На этом этапе необходимо получить опорные координаты для каждого помещения:

$$Room_i = ([P_1, Name_1], [P_2, Name_2] \dots, [P_m, Name_m]).$$

То есть для каждой комнаты необходимо хранить все сигналы видимых точек доступа (P_j) и имена этих точек ($Name_j$).

Далее необходимо сравнить список видимых точек со списком разрешённых, чтобы отфильтровать неразрешённые:

$$Rm_i = allowed\{ Room_i \}.$$

Далее остаётся лишь сохранить в базу данных информацию о помещении, содержащую информацию о пяти наиболее сильных источниках для данного помещения:

$$DB_{Room_i} = mazZ_1^M \{ Rm_i \}.$$

По умолчанию выбирается $Z = 5$, то есть берутся 5 наиболее мощных точек доступа. Однако этот параметр трудно задать фиксировано, так как он зависит от многих факторов. Например, в спортзале желательно сохранять 7 точек доступа, а в учебных помещениях с большим количеством *Wi-Fi* точек доступа будет достаточно хранить 3. Основной этап работы системы представлен на блок-схеме, рисунок 1. Пронумерованные блоки рисунка 1:

1) Устройство снимает уровни сигнала до опорных точек и идентифицирует эти точки ($Room_i$).

2) Отбрасываются точки с низким уровнем и проверяется, как много точек доступа имеют достаточный уровень сигнала (Rm_i). Это действие выполняется 3–5 раз за короткий промежуток времени. Сильно отличающиеся от других результаты эксперимента отбрасываются, среди оставшихся одноимённые координаты усредняются.

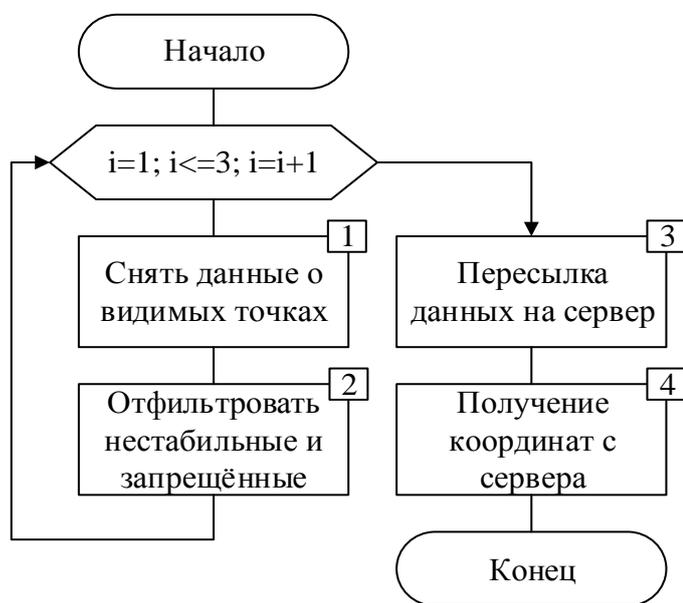


Рисунок 1 – Блок-схема работы клиента

3) Пересылка координат на сервер, согласно графику расписания звонков. Кроме статически указанных интервалов поддерживаются и динамические временные интервалы (гибкий график).

4) Сервер обрабатывает полученные данные и определяет, в каком конкретно помещении находится данный объект. Статистика хранится на сервере для последующего использования.

Слабым местом методики является необходимость переобучения системы в случае реконфигурации оборудования. Например, изменение типа или местоположения точек доступа, добавление новых точек с высоким уровнем сигнала и тому подобное. Таким образом, после каждой реконфигурации требуется корректировка информации для позиционирования. В некоторых случаях это может быть эквивалентно первоначальному обучению системы [2]. Для решения данной проблемы предлагаем использовать так называемый динамический эталон.

Кроме непосредственно самой методики следует учесть несколько дополнений к ней.

1. Необходимо делать три контрольных измерения. Если значения хотя бы двух измерений совпали, то положение найдено. Это отображено в блоке 1 на рисунке 1.

2. При неудачном измерении координат в текущий момент времени по трём измерениям необходимо информировать сервер статистики о том, что положение в данный момент времени не известно.

3. Имеет смысл помнить о неточностях алгоритма – если стоять в аудитории возле стены, то в качестве координат можно получить координаты соседнего помещения.

4. Период снятия данных должен быть выставлен в соответствии с расписанием занятий: во время пар снимать данные раз в полчаса, во время перемен – раз в пять минут.

После введения в методику приведенных уточнений были получены результаты, показанные на рисунке 2. Каждые 1000 секунд производилось 3 измерения текущей координаты, которые затем округлялись по точности до помещения. Если хотя бы 2 измерения из 3 совпадают, то помещение считается корректно найденным.

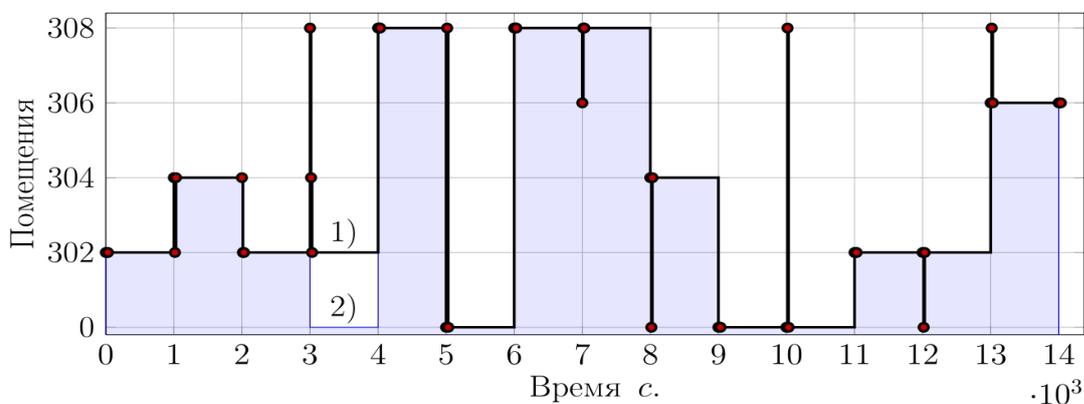


Рисунок 2 – График положения от времени:
1) вычисленное положение; 2) реальное положение

В данной работе предложена методика позиционирования внутри помещения на основании существующей *Wi-Fi* сети. Применение данной методики позволяет оперативно получить информацию о положении студента или преподавателя в ВУЗе (используя программное обеспечение сервера, который собирает статистику), а также вести автоматизированный учёт посещения занятий для модульно-рейтинговой системы.

Литература

1. Храбров, Д.Е. Методика позиционирования и контроля посещаемости студентов на основании WiFi-сети университета / Д.Е. Храбров, И.А. Мурашко / М-во образования Респ. Беларусь, Бел. гос. ун-т. инф-ки и рад-ки. – Минск, 2014. – С. 254–255.

2. A survey of mathematical methods for indoor localization / F. Seco, A.R. Jimenez, C. Prieto et al. // Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on. – 2009. – P. 9–14.

А.Н. Цябус (УО «ГГУ им. Ф. Скорины», Гомель)
Науч.рук. **Г.Л. Карасёва**, канд. физ.-мат. наук, доцент

ИСПОЛЬЗОВАНИЕ UNITY3D ДЛЯ СОЗДАНИЯ ИГРОВОГО ПРИЛОЖЕНИЯ

Unity3d является современным кроссплатформенным движком для создания игр и приложений. Разработкой занимались Unity Technologies. С помощью данного движка можно разрабатывать не только приложения для компьютеров, но и для мобильных устройств (например, на базе Android, IOS), игровых приставок и других девайсов.

К основным характеристикам движка стоит отнести то, что в среде разработки Unity интегрирован игровой движок, иными словами, присутствует возможность протестировать игровое приложение не выходя из редактора. Во-вторых, Unity поддерживает импорт огромного количества различных форматов, что позволяет разработчику игры конструировать сами модели в более удобном приложении, а Unity использовать по прямому назначению – разрабатывать продукт. В-третьих, написание сценариев (скриптов) осуществляется на наиболее популярных языках программирования – C# и JavaScript.

Unity – это мультиплатформенный инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows и OS X. Созданные с помощью Unity приложения