

# Мобильные Bluetooth теги

Намиот Д.Е.

**Аннотация**—В статье рассматриваются вопросы разработки мобильных сервисов, связанных с беспроводными тегами. В первой части работы анализируются устройства (теги) на основе Bluetooth Low Energy и предложенная компанией Apple технология iBeacon. Остальная часть статьи посвящена развитию модели персональных и мобильных тегов на основе Bluetooth. Основным достижением этой модели является то, что она охватывает все этапы жизненного цикла – представление тега (тегов), привязка к ним информационного наполнения и создание сервисов на основе имеющихся тегов. В качестве тегов могут выступать как мобильные телефоны, так и существующие устройства с поддержкой Bluetooth. Как области применения данного подхода можно назвать приложения для торговых и сервисных организаций, навигацию в помещениях, а также контекстно-зависимые приложения для Smart Cities.

**Ключевые слова**—Bluetooth, iBeacon, network proximity, context-aware

## I. ВВЕДЕНИЕ

Настоящая работа является развитием идей, впервые изложенных в статье [1].

Под термином тег (RFID tag, NFC tag и т.д.) здесь понимается некоторое аппаратное устройство, которое поддерживает определенный коммуникационный протокол (протоколы). Любая подобная система состоит из считывающего устройства и меток-передатчиков (транспондеров). Устройства могут пассивные (не имеют источника питания и опрашиваются считывателем) и активные. В качестве транспондера могут выступать различные устройства. В данной работе в качестве считывателей рассматриваются мобильные телефоны.

Применение у тегов может быть самое различное. Например, RFID теги (распространяемый ими сигнал) могут использоваться для идентификации. Для NFC устройств, стандарт описывает свои протоколы и форматы обмена данными, которые могут использоваться, например, для организации платежей или загрузки какой-либо информации на другие устройства.

Одним из широко поддерживаемых коммуникационных стеков в таких устройствах является Bluetooth. Это связано с технологичностью производства таких устройств, с историей самого

протокола, его достаточной проработанностью и т.д. До широкого распространения Wi-Fi, Bluetooth (приложения на базе Bluetooth), по сути, являлся основным способом для связи мобильных устройств.

В последнее время большое внимание привлекает к себе технология iBeacon, анонсированная компанией Apple в составе iOS 7. Эта технология основана на сравнительно новом стандарте Bluetooth Low Energy (BLE) [2]. Стандарт BLE был специально разработан для использования в сенсорах, которые долгое время должны работать без замены батареи. iBeacon от Apple представляет собой как раз пример такого сенсора [3]. Идея использования состоит в том, что приложение (сервис) может определять наличие подобного рода тегов поблизости и, в зависимости от этого, выполнять какие-либо действия (получать какую либо информацию), исходя из предположения, что телефон (приемник), на котором работает данное приложение, находится поблизости от конкретных тегов. Каждый тег при этом, естественно, должен как-то идентифицироваться. И именно эта идентификация тегов и учитывается приложениями. Приложения, по сути, привязаны к некоторой группе тегов, и наоборот, группа тегов обслуживает конкретное приложение.

В настоящее время, Apple не является единственной компанией, которая производит подобные устройства [4]. Но интересным моментом является то, что поддержка BLE осуществляется на уровне операционной системы iOS. В качестве тега BLE могут выступать сами мобильные телефоны. Также, установленное однажды приложение продолжает принимать идентификаторы тегов, даже не будучи непосредственно запущенным.

Вместе с тем, BLE пока не получил широкого распространения, то есть, по крайней мере, не заменил и не вытеснил традиционные Bluetooth устройства. Поддержка Bluetooth уже давно существует практически во всех мобильных телефонах. В настоящей работе рассматривается программная модель, которая позволяет создавать Bluetooth теги непосредственно на мобильном телефоне. Основным достижением этой модели является то, что она охватывает весь жизненный цикл – создание тега (тегов), привязка к ним информационного наполнения, создание сервисов на основе имеющихся тегов, а также их эксплуатация. При этом, собственно беспроводной протокол Bluetooth используется для целей определения сетевой близости и может быть прозрачно для сервисов заменен в какой-то момент времени на BLE.

Работа структурирована следующим образом. В разделе II рассматривается работа iBeacons. В разделе

Статья получена 20 апреля 2014.

Д.Е. Намиот – старший научный сотрудник факультета ВМК МГУ им. М.В. Ломоносова (e-mail: dnamiot@gmail.com).

III рассматривается предложенная модель Bluetooth тегов. Раздел IV посвящен описанию возможных сервисов и направлениям развития системы.

Представленные в данной работе результаты получены в рамках исследований, проводящихся в лаборатории ОИТ факультета ВМК МГУ им. М.В. Ломоносова [5]. Их непосредственное применение возможно, например, в рамках проектов по Internet of Things [6], телекоммуникационных сервисах для geofence [7] и т.п.

## II. iBEACON

В данном разделе мы хотели бы кратко изложить принципы работы сервисов на базе технологии iBeacon. Общая схема работы проиллюстрирована на рисунке 1.

Мобильный абонент проходит мимо расставленных тегов. Его телефон (приложение в телефоне) принимает идентификацию видимых в данный момент тегов.

Возможно автоматическое проведение оценочных расчетов, основанных на силе сигнала и выдающих для каждого видимого тега качественную оценку: далеко, близко, рядом и т.п.

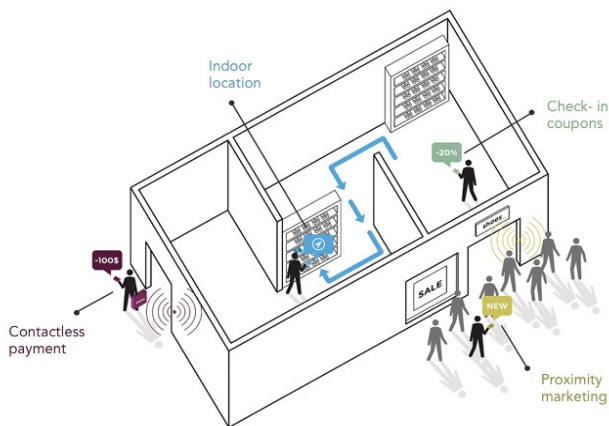


Рисунок 1. iBeacon [8]

Отметим, что, в принципе, в качестве тега может выступать и другое мобильное устройство (телефон), оснащенное BLE сенсором.

Технически, iBeacon – это все же не устройство, а протокол, работающий поверх BLE. По сути, все, что делает iBeacon – это широковещательная рассылка с постоянным временным интервалом (раз в секунду) собственного идентификатора (Рис. 2).



Рисунок 2. Рассылка сообщений [9]

iBeacon просто постоянно сообщает о своем присутствии. Собственно говоря, low energy profile и нужен как раз для долговременного обеспечения этого самого постоянства.

Никаких данных iBeacon не посылает. В рассылке участвует только некоторый уникальный идентификатор устройства (UUID) и два числа, обозначаемые в спецификации как “major” и “minor”. Каждое возможное приложение будет использовать (слушать, получать) один или несколько из заданных (определенных) в нем UUID. При этом приложение будет получать информацию только о специфицированных (заданных) в нем самом UUIDs. Получить информацию обо всех транслируемых в данной области UUID нельзя. А два дополнительных значения для каждого UUID (“major” и “minor”) каждое приложение может интерпретировать самостоятельно. Их интерпретация – это уже часть алгоритма приложения. Например, для идентификации товаров в магазине можно предположить, что одно из них определяет торговый зал (секцию), а второе – полку с товаром. И так далее. Это может оказаться какой-то идентификацией объекта, предмета – все, что может потребовать идентификации.

Приложение (сервис) прослушивает эфир в поисках доступных UUID, определяет их дополнительные номера, а далее использует эту информацию в своих алгоритмах. Прослушивая UUID, исполнительная система может также оценивать расстояние до соответствующего UUID (от 50 до 30 м – рисунок 3):

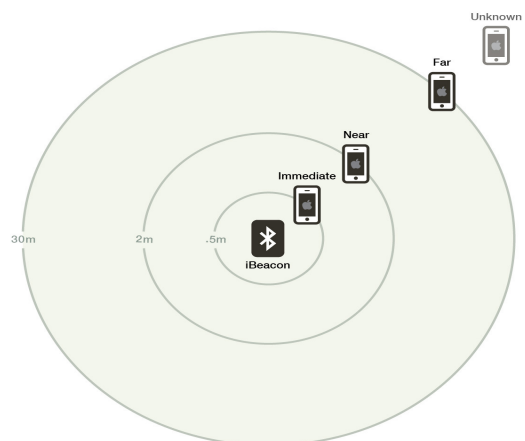


Рисунок 3. iBeacon расстояния [9]

Возможные категории: Immediate (до 0.5 м), Near (до 2 м), Far (до 30 м), Unknown (больше 30 м).

Опросом силы сигнала дело Гн ограничивается. Производители тегов (Qualcomm, в частности) предлагают и другие интерфейсные API, которые позволяют, например, опрашивать уровень заряда батареи тега и его температуру. Теги от Estimote позволяют получить доступ к данным акселерометра.

Очевидно, что необходимо как-то подготовить устройства (установить UUID, major и minor), а также

синхронизовать эти данные со своим приложением. Напомним, что оно работает только лишь с предписанными ему UUID [10]. Это довольно тонкий момент во всей технологической цепочке, который может затруднить разработку приложений. И это один из моментов, на котором будет акцентирована предлагаемая ниже новая модель. Одним из основных ее достоинств будет поддержка полного жизненного цикла – от создания сервисов до их эксплуатации (сбора статистики и т.п.).

По нашему мнению, именно “локальность” iBeacon тегов может являться основной проблемой. Если iBeacons используются для уведомления, например, покупателей в магазине, то покупателю нужно будет загружать отдельное приложение для каждого магазина. Что представляется весьма сомнительным и, к тому же, отсекает всю армию разовых посетителей. А единое приложение можно представить только для какой-то большой торговой сети, которая централизованно ведет управление запасами и их расположением на торговых полках для всех своих подразделений.

Другой вопрос связан уже с представлением информации, полученной на основе найденных тегов. В качестве базового метода рассматривается рассылка уведомлений (push notifications). Технически, это объяснимое решение. Уведомления изначально были спроектированы для работы с установленными, но необязательно запущенными приложениями. Однако с точки зрения пользователя это может оказаться совсем не так уж удобно. Представьте себе достаточно большой магазин, в котором в каждый момент времени существуют сотни специальных предложений. Трудно представить себе полезную ситуацию, когда все они переводятся в уведомления. А с другой стороны, если предложений мало – то они вполне могут быть представлены в виде объявлений в самом магазине (на ценниках товаров и т.д.), то есть, быть доступны без мобильных приложений.

А с другой стороны, разграничение видимости для тегов существует только на уровне приложений. На уровне операционной системы производителю доступна информация обо всех вещающих в данной области тегах. Это может привести к интересному результату. Известен и используется подход, связанный с пассивным мониторингом мобильных устройств [11]. При пассивном мониторинге используется анализ беспроводных протоколов (сигналов, посылаемых телефоном) и, соответственно, он не требует загрузки от пользователя специальных программ или выполнения каких-либо действий в приложениях (check-in). Здесь также возможны решения с доставкой уведомлений мобильным абонентам, находящимся в некоторой выделенной области [12].

Но ведь если мобильная ОС, так или иначе, регистрирует сигналы от тегов, то это также может быть отнесено к пассивному мониторингу (в смысле отсутствия требований к наличию приложения). Производитель мобильной ОС становится регистратором пользовательских перемещений. И в этом

смысле, Apple может оказаться главным бенефициаром от повсеместного внедрения iBeacons тегов.

### III. BLUETOOTH DATA POINTS

Наш подход к созданию персональных мобильных Bluetooth тегов основан на идеях использования сетевой близости, описанных в проекте SpotEx. Идея состоит в создании связи некоторого задаваемого пользователем контента и идентификации сетевых узлов [13].

В SpotEx пользователь (издатель) может связать некоторый контент (практически – произвольный код на HTML или просто ссылку на веб-ресурс) с доступными идентификаторами точек доступа Wi-Fi. Точки доступа могут быть как существующими и публично доступными, так и специально созданными в рамках данного проекта [14]. В частности, они могут быть открыты (запущены) непосредственно на мобильном телефоне (телефонах) [15].

Привязка контента к сетевым узлам означает задание правил (предикатов) видимости. Например, если точка доступа с SSID Café видна в данный момент времени мобильному пользователю, то ему становится доступным (открывается, показывается) некоторый соответствующий этому факту контент. Правила активации (видимости) определяет непосредственно автор (владелец) контента. Для читателя (мобильного абонента) конечное приложение выглядит как браузер. Приложение оценивает доступные сетевые узлы, определяет сработавшие правила и предьявляет для просмотра соответствующий набор данных [16]. На практике это выглядит как формирование динамической веб-страницы, содержание которой формируется из отдельных фрагментов, соответствующих сработавшим правилам.

Отметим, что речь при этом не идет о присоединении к какой-либо точке доступа Wi-Fi. Контент может располагаться где угодно в сети. Точка доступа Wi-Fi (ее атрибуты, типа наименования и силы сигнала) работает просто как триггер. Если мобильное приложение “видит” данную точку доступа, то телефон (абонент) где-то поблизости. В этом основная идея. И “браузер” становится средством доступа к локальному контенту. Это принципиальное отличие от модели представления данных, построенной целиком на уведомлениях. В “браузере” просмотр данных инициируется мобильным абонентом, а не системой.

Теперь мы хотим перенести эту же самую схему для работы с Bluetooth. В принципе, эта возможность уже обсуждалась в работах, посвященных SpotEx.

Для Bluetooth картина использования сетевой близости концептуально одинакова, но есть и важные технические различия. Они касаются расстояния, на котором можно обнаруживать Bluetooth точки (теги активации данных) – это, в целом, будет меньше, чем для Wi-Fi. Определение (поиск) Bluetooth узлов, находящихся в так называемом discoverable mode работает медленнее, чем поиск Wi-Fi узлов. С другой стороны, количество Bluetooth узлов, используемых в качестве точек привязки данных, будет (опять в

среднем) больше, чем Wi-Fi узлов. И, наконец, важный технический момент - Bluetooth точку можно запустить программно. Иными словами, приложение на мобильном телефоне может программно запустить (остановить) Bluetooth узел в так называемом discoverable mode. Это режим, при котором Bluetooth узел виден при сканировании другим Bluetooth узлом. Подчеркнем, что речь не идет об установлении соединения и какой-либо передаче данных. Все опирается просто на сам факт видимости беспроводной точки.

В целом, схема работы может быть проиллюстрирована на следующих рисунках. Все начинается с того, что один из пользователей – издатель, создает некоторый контент и связывает его с идентификацией своего телефона. Или использует какой-то существующий контент, ассоциируя его со своим мобильным устройством (рисунок 4).

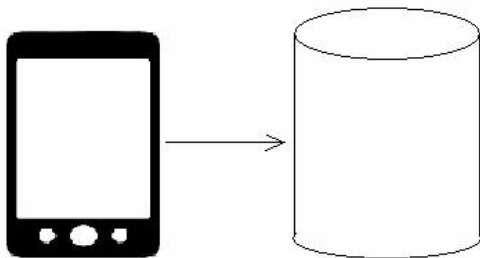


Рисунок 4. Создание контента

На следующем шаге издатель включает точку Bluetooth на своем мобильном устройстве (рисунок 5).

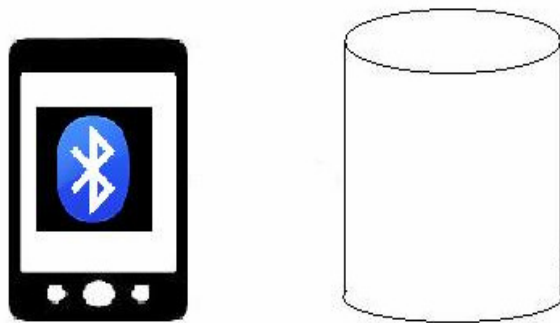


Рисунок 5. Создание точки Bluetooth

Очевидно, что эти два шага могут оказаться и разделенными по времени. Можно разово создать какой-то контент, а затем многократно активировать (деактивировать) “включающую” (“выключающую”) его точку Bluetooth. Можно добавлять (удалять, редактировать) контент, относящийся к какой-либо точке доступа и т.д.

На последнем шаге мобильный пользователь (читатель) запускает приложение-браузер (рисунок 6). Приложение сканирует видимые точки доступа, определяет их атрибуты и использует эту информацию для запроса активного контента, ассоциированного с

доступными сетевыми узлами.

Контент хранится вне мобильного устройства. С помощью мобильного устройства он только создается (редактируется).

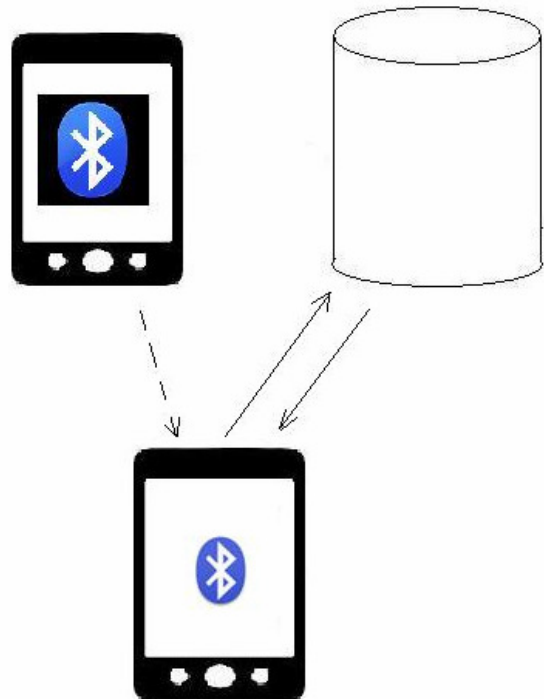


Рисунок 6. Получение контента

В качестве контента мы можем рассматривать любой фрагмент HTML кода. В частности, просто ссылка в самом простейшем случае. “Браузер” динамически компонует выходную страницу из отдельных фрагментов и показывает ее пользователю.

Здесь возникает интересная задача об использовании информации о доступных точках Bluetooth непосредственно в описании веб-страниц [17]. В указанной работе речь идет о том, чтобы в тегах *div* элементов представлять те же самые правила доступности данных. Это может быть обеспечено с помощью так называемых HTML5 custom attributes. Соответственно, обработка этих атрибутов позволит просто делать соответствующие фрагменты видимыми (невидимыми). Это позволило бы использовать уже обычный браузер для просмотра такого контента.

Заметим, что весь процесс может быть полностью анонимным. Как и в случае SpotEx, речь не идет о соединении по Bluetooth. Узлы Bluetooth выступают только в роли триггеров, которые обеспечивают срабатывание правил для представления данных.

Подобного рода система может быть названа Bluetooth Data Points (BDP).

Относительно модели данных, поддерживаемой в системе, можно представить следующий набор реляционных таблиц.

MAC - информация по MAC-адресам Bluetooth точек  
(*Id* – primary key  
*address* – содержит MAC-адрес точки  
*start* – время первой регистрации в системе )

Announces – собственно объявления  
(*Id* – primary key,  
*start* – время создания  
*modified* – время последней редакции  
*status* – описание текущего состояния (например, активно или нет)  
*Id\_MAC* – foreign key. Ссылка на MAC-адрес  
*Content* – содержание объявления)

PointsLog – журнал для регистрации операций с точками доступа  
(*Id* – primary key  
*Id\_MAC* – foreign key. Ссылка на MAC-адрес  
*ts* – время операции  
*lat* – широта места (опционально)  
*lng* – долгота места (опционально)  
*action* – описание действия. 1 – запущена, 0 – остановлена)

Log – журнал запросов сканера  
(*Id* – primary key  
*ts* – дата и время запроса,  
*user-mac* – адрес пользователя в сканере)

LogData – результаты запросов  
(*Id\_log* – foreign key, ссылка на запрос  
*Id\_announce* – foreign key, ссылка на объявление )

Структура может меняться. Данный вариант соответствует текущему состоянию разработки. Что, например, может быть расширено? Может быть расширено описание контента. Например, явно введены типы для контента (веб-ссылка, ссылка на профайл в социальной сети, некоторый текст). Могут появиться категории (ключевые слова) для контента. Тогда, соответственно, браузер получит возможность группировать (фильтровать) контент по категориям.

Наличие журнала запросов необходимо для того, чтобы обеспечить пользователям статистику по их объявлениям: сколько раз оно было показано за заданный временной промежуток. Это важно, в первую очередь, для сервисов на уровне предприятия.

К одной точке может быть, потенциально привязано несколько объявлений. Автор объявлений должен иметь возможность активировать (деактивировать) их в произвольном порядке. Все активные объявления будут доступны для браузера, если включена соответствующая точка доступа.

Операции с точками Bluetooth регистрируются опять-таки для возможных сервисов статистики. Поскольку точка привязки данных может быть мобильной, может оказаться интересным сохранять истинные географические координаты, где была выполнена та или иная операция. В частности, это даст возможность

строить гео-информационные системы. Например, показывать на карте места, где именно в данный момент есть какие-либо местные объявления.

Другим возможным направлением развития будет являться введение аккаунтов для пользователей, публикующих (редактирующих) данные. Это не нужно для использования подобного рода сервисов в персональных целях, но будет необходимо для поддержки сервисов на уровне предприятий.

#### IV. ПРИЛОЖЕНИЯ И РАЗВИТИЕ

Из возможных приложений подобного рода сервиса мы можем назвать любые проекты, которые связаны с распространением гипер-локальных данных. Данные, которые имеют смысл в некотором локальном контексте. Например, для торгового центра или заведения общественного питания – распространение специальных предложений. Это соответствует тому, что мы описывали для SpotEx [18]. Другая интересная область – это приложения для Smart Cities. Подобный подход позволяет, например, уйти в описании городских сервисов от гео-привязки и перейти к гораздо более точной привязке описаний к месту расположения сетевых устройств. При этом предложенный способ будет работать и помещениях, там, где доступ к гео-позиционированию может быть затруднен.

Наличие триггеров непосредственно на мобильном устройстве, позволяет включать их в произвольном месте. Здесь мы снова следуем модели динамических LBS [19]. Фактически, данные следуют за мобильным устройством и оказываются доступными в месте фактического присутствия мобильного телефона (издателя).

Модель включает в себя сбор статистики по обращениям к контенту. Можно собирать информацию о месте включения (выключения) каждой точки, данные по обращениям к конкретным данным (просмотры).

Ориентация на мобильные устройства никак не закрывает возможность использовать точки доступа Bluetooth, например, на персональных и переносных компьютерах. Точки доступа Bluetooth также присутствуют на элементах ИТ-инфраструктуры (например, принтеры), в приборных панелях автомобилей и т.д. С использованием точек Bluetooth в автомобилях (то есть, с привязкой какого-либо контента к атрибутам точки доступа в автомобиле), идея с динамическими LBS становится наиболее наглядной.

Отличие от модели iBeacon состоит в том, что у издателя остается возможность динамического включения (выключения) вещателей. Другое отличие состоит в том, что в такой модели каждое приложение “видит” всех вещателей. Уже само приложение будет решать, интересна ему конкретная информация или нет. Другой особенностью является то, что поддержка собственно конфигурирования системы является неотъемлемой частью модели.

Как сигнал мобильным абонентам о том, что в данной области есть, возможно, интересный им контент, можно использовать какой-либо видимый знак (наклейку). По аналогии с тем, как помечаются области с поддержкой

Wi-Fi. Кстати, вполне возможно использовать здесь стандартную иконку для Bluetooth.

Другой интересный момент состоит в том, что описанная модель может, в принципе, быть реализованной и на базе BLE. Что является основным в данной модели? Во-первых, это возможность динамически привязать свой собственный контент к элементам сетевой инфраструктуры (тегам). Во-вторых, это возможность динамически определять видимость тех или иных данных. В изложении выше это определяется возможностью запуска Bluetooth точки, в так называемом режиме опознавания (discovery mode). Это делает точку видимой для программ, которые ищут окрестные Bluetooth устройства. Соответственно, для одной из таких программ (“браузера”) становится видны атрибуты Bluetooth точки (ее MAC-адрес), а уже по этим атрибутам и можно найти связанный с ней контент. Для BLE это, очевидно, должно реализоваться посредством запуска программного вещателя на своем телефоне.

## V. ЗАКЛЮЧЕНИЕ

В первой части данной работы анализируются устройства (теги) на основе Bluetooth Low Energy и предложенная компанией Apple технология iBeacon. В основной части работы представлено развитие модели персональных мобильных Bluetooth тегов Bluetooth Data Points (BDP). Основным достижением этой модели является поддержка всех этапов жизненного цикла приложений (сервисов) на основе тегов – создание вещателя (вещателей), привязка к ним информационного наполнения и создание сервисов на основе имеющихся тегов. Для создания и эксплуатации сервисов на основе такой модели достаточно одних мобильных телефонов. В качестве возможных областей применения данной модели можно назвать контекстно-зависимые приложения для Smart Cities, приложения для навигации в помещениях, а также приложения для торговых и сервисных организаций.

## БИБЛИОГРАФИЯ

- [1] Намиот Д. Е. Персональные Bluetooth теги //International Journal of Open Information Technologies. – 2014. – Т. 2. – №. 3. – С. 35-39.
- [2] Gomez, Carles, Joaquim Oller, and Josep Paradells. "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology." *Sensors* 12.9 (2012): 11734-11753.
- [3] Dilger D. E. Inside iOS 7: iBeacons enhance apps' location awareness via Bluetooth LE //ed: Apple Insider. – 2013.
- [4] Lerner, T. (2013). International Comparisons. In *Mobile Payment* (pp. 137-142). Springer Fachmedien Wiesbaden.
- [5] Намиот Д., Сухомлин В. О проектах лаборатории ОИТ //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 5. – С. 18-21.
- [6] А.А. Волков, Д.Е. Намиот, М.А. Шнепс-Шнеппе. О задачах создания эффективной инфраструктуры среды обитания //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 7. – С. 1-10.
- [7] Namiot, D. (2013). GeoFence services. *International Journal of Open Information Technologies*, 1(9), 30-33.
- [8] iBeacon <http://gigaom.com/2013/09/10/with-ibeacon-apple-is-going-to-dump-on-nfc-and-embrace-the-internet-of-things/> Retrieved: Feb, 2014
- [9] iBeacon experiments <http://blog.nerdery.com/2013/11/nerdery-labs-ibeacon-experiments/> Retrieved: Feb, 2014

- [10] Proximity Kit <http://www.radiusnetworks.com/proximity-kit.html> Retrieved: Feb, 2014
- [11] Sneps-Sneppé, M., & Namiot, D. (2013). Spotique: A New Approach to Local Messaging. In *Wired/Wireless Internet Communication* (pp. 192-203). Springer Berlin Heidelberg.
- [12] Namiot, D., & Sneps-Sneppé, M. (2013). Geofence and Network Proximity. In *Internet of Things, Smart Spaces, and Next Generation Networking* (pp. 117-127). Springer Berlin Heidelberg.
- [13] Namiot, D. (2013). Network Proximity on Practice: Context-aware Applications and Wi-Fi Proximity. *International Journal of Open Information Technologies*, 1(3), 1-4.
- [14] Namiot, D., & Sneps-Sneppé, M. (2012, May). Wi-Fi proximity as a Service. In *SMART 2012, The First International Conference on Smart Systems, Devices and Technologies* (pp. 62-68).
- [15] Namiot, D., & Sneps-Sneppé, M. (2012, April). Wi-Fi Proximity and Context-aware Browsing. In *ICDT 2012, The Seventh International Conference on Digital Telecommunications* (pp. 1-7).
- [16] Namiot, D., & Sneps-Sneppé, M. (2012, April). Proximity as a service. In *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on* (pp. 199-205). IEEE.
- [17] Намиот Д. Е. Локальные веб-страницы //International Journal of Open Information Technologies. – 2014. – Т. 2. – №. 2. – С. 29-33.
- [18] D. Namiot and M. Sneps-Sneppé "Context-aware data discovery", *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on*, pp. 134-141, DOI: 10.1109/ICIN.2012.6376016.
- [19] José, R., Moreira, A., Rodrigues, H., & Davies, N. (2003). The AROUND architecture for dynamic location-based services. *Mobile Networks and Applications*, 8(4), 377-387.

# On mobile Bluetooth tags

Namiot D.E.

**Abstract—** The article examines the development of mobile services related to wireless tags. The first part analyzes the devices (tags) based on Bluetooth Low Energy and iBeacon technology offered by Apple. The rest of the paper is devoted to the development of our model of personal and mobile tags based on Bluetooth. The main achievement of this model is that it covers all stages of the life cycle: how to present our tag (tags), how to bind some context to them and how to use tags in context-aware services. As mobile tags we can use mobile phones as well as existing devices with Bluetooth support. As the use cases for this approach we can mention context-aware applications for Smart Cities, indoors navigation, as well as applications for retail.

**Keywords—** Bluetooth, iBeacon, network proximity, context-aware.