

О телекоммуникационных сервисах

Гурьев Д.Е., Намиот Д.Е., Шнепс М.А.

Аннотация—В статье рассматриваются вопросы разработки телекоммуникационных сервисов, анализируется опыт их создания и внедрения. Вопросы, связанные с разработкой и эксплуатацией подобного рода сервисов явились основой для определения тем многих квалификационных работ (магистерских диссертаций, дипломов), выполненных выпускниками МГУ в лаборатории Открытых Информационных Технологий факультета ВМК МГУ. В статье анализируются полученные результаты, приводятся примеры квалификационных работ и перспективные направления для исследования.

Ключевые слова—телеком, сервисы, голосовые услуги, мобильные сервисы.

I. ВВЕДЕНИЕ

В работе рассматриваются вопросы, связанные с проектированием, разработкой и эксплуатацией телекоммуникационных сервисов. В некоторой степени мотивационным фактором для написания данной работы явилась статья [1], в которой описывались результаты и темы для квалификационных работ (магистерских диссертаций, дипломных работ), выполнявшихся (предлагающихся к исследованию) в лаборатории Открытых Информационных Технологий факультета ВМК МГУ им. М.В. Ломоносова. Мы можем отметить, что немало работ за прошедшее время было посвящено именно телекоммуникационным сервисам. Соответственно, настоящая статья представляет собой попытку систематизации этих усилий.

Не претендуя на полноту охвата всей картины в отдельно взятой статье, все же необходимо отметить, что разработка сервисов в отечественных публикациях и учебных курсах высшей школы в известной степени выпала из рассмотрения. Если говорить точнее, то практически не рассматривается. Научные работы, традиционно относящиеся к телекоммуникационной тематике, как правило, охватывают лишь описание протоколов взаимодействия. В терминах семиуровневой сетевой модели – относятся к нижним уровням. Все это выглядит так, как будто все усилия посвящены именно созданию телекоммуникационной аппаратуры, которая, тем не менее, в отечественном варианте практически отсутствует [2].

Статья получена 15 февраля 2014.

Д.Е.Гурьев - научный сотрудник факультета ВМК МГУ им. М.В. Ломоносова (e-mail: gouriev@icenet.ru)

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

М.А. Шнепс – ведущий научный сотрудник ЦНИИС (email: sneps@mail.ru).

Все это приводит к тому, что целые направления выпадают из какого-либо рассмотрения. Например, что выпускники будут знать о текущих проектах в Internet of Things [3], европейских проектах FI-WARE и т.д.?

И дело здесь не только в том, что данная статья написана для журнала, издающегося на факультете, относящемся к математике и компьютерным наукам, а не к связи. Анализ всех последних проектов, относящихся, например, к упомянутому выше направлению IoT указывает на то, что это связано, в первую очередь именно с программированием. Конечное выражение всего, видимая часть – это именно сервисы. Программы, которые их реализуют. Анализ и разработка протоколов связи относится именно к разработке аппаратуры, а системная архитектура, например, это облачные технологии, распределенные системы, обработка больших данных и т.д.

В этой статье мы остановимся именно на голосовых сервисах. Традиционная для телефонии часть – звонок. То, с чего все начиналось. И то, чем все продолжается. Заметим, что при этом не важно, как именно телефонный звонок осуществляется – по традиционным каналам или через Voice Over IP.

II. ИДЕОЛОГИЯ СЕРВИСОВ

Основная идея в построении сервисов, основанных на голосовой связи, состоит в том, что они должны заключаться в каком-либо инкорпорировании (встраивании) звонков в другие приложения. Собственно говоря, это же относится и к другим телекоммуникационным сервисам. Не существует (и не будет существовать) никакого “золотого” сервиса (приложения), которое разом решит все проблемы абонентов, раз и навсегда снабдив оператора постоянно растущим доходом. Другим важным моментом является признание того, что установка платной услуги (услуги с повышенной оплатой) не является никаким сервисом. И не решит проблем ни пользователей, ни операторов. Время премиальных услуг (если такое и было когда-то) уже ушло. Звонок – это базовая услуга. Ничем премиальным здесь просто не будут пользоваться (ну или не будут пользоваться в коммерчески значимом объеме). Вместе с тем, телекоммуникационные операторы вполне успешно собирают деньги именно за базовые услуги. И целью сервисов, по нашему убеждению, является именно помощь в этом. Бизнес-модель сервисов должна заключаться именно в увеличении потребления базовой услуги. Которую операторы хорошо умеют обчислять и за которую вполне успешно собирают деньги.

Встраивание обработки звонков в сторонние приложения как раз и приводит к тому, что звонок (голосовое общение) становится еще одним из каналов взаимодействия с приложением. И при этом такое взаимодействие естественным образом увеличивает потребление базовой телекоммуникационной услуги.

Звонок, встроенный в приложение, означает, естественно, программную обработку этого звонка. Что под этой обработкой подразумевается, и какие возможности она включает? Программная обработка – это всегда возможность запустить на исполнение свой собственный код (функцию, процедуру, модуль и т.д.) по факту поступления звонка на некоторый номер. Технически, необходимо совсем немного функций, комбинация которых и дает возможности строить сервисы. Таковых базовых функций мы можем насчитать шесть.

Во-первых, звонок необходимо совершать. Собственный программный код должен обладать возможностью позвонить по заданному номеру и выполнить какие-либо действия по факту приема звонка или отсутствию ответа.

Во-вторых, звонок необходимо просто завершать. Программно необходимо звонок принять, то есть, запустить на исполнение некоторый собственный код по факту поступления входящего звонка, в этом коде прочитать (определить) номер позвонившего абонента и завершить звонок. Пример использования – любой сервис голосования по телефону. Код, который запускается по факту звонка, например, просто увеличивает счетчик в некоторой базе данных. А номер абонента нужен, чтобы предотвращать повторные голосования. С практической точки зрения, это именно завершение вызова. Программа-обработчик обновляет базу данных со статистикой голосования и просто завершает звонок (“кладет трубку”).

Следующая функция – проигрывание некоторого звукового файла. Типичный пример – автоответчик. Программа, которая принимает звонок (то есть, “снимает трубку”) определяет, какой аудио-файл является подходящим. Для этого, например, программа может руководствоваться номером позвонившего, временем дня, погодой и другими доступными данными. Контекстом, например, как контекстно-зависимые приложения [5]. После чего проигрывает (“дает прослушать”) позвонившему абоненту выбранный аудио-файл. Аналогично, функция должна работать для совершения звонка. Позвонить, дождаться ответа и проиграть аудио-файл.

Следующий момент – это переадресация. Принятый звонок будет переадресован на какой-то другой номер. Программа принимает звонок, по какому-то своему алгоритму определяет новый номер и переадресует звонившего абонента на новый номер. Примеры сервисов – виртуальный номер.

Следующая функция – это запись аудио-файлов. Программа принимает звонок (“снимает трубку”) и записывает все сказанное позвонившим абонентом (или позвонившей программой). Пример сервиса – голосовая почта.

И последняя возможность – это взаимодействие с пользователем. Прием тоновых посылок. То, что позвонивший абонент (другая программа) указывает посредством дополнительного набора.

Рассмотрим пример сервиса (реально реализовывался на платформе от Аудиотеле). Система подачи объявлений (classified), где в качестве контактной информации пользователь (автор объявления) указывает некоторый код для связи. Код этот привязан к номеру мобильного телефона и может быть получен посредством регистрации телефона на веб-портале проекта или же посредством телефонного звонка на некоторый сервисный номер. В любом случае, код (цифровая последовательность) привязан к телефонному номеру. Тип отношения – многие к одному. То есть одному телефонному номеру может соответствовать несколько кодов. При публикации объявления автор указывает один его кодов для связи.

Если читателю объявления необходимо связаться с автором (его заинтересовало авторское объявление), то он просто звонит на сервисный номер и указывает тоновыми послылками код автора. Этот код не нужно запоминать – он указан непосредственно в объявлении.

После указания кода читатель может записать голосовое сообщение для автора объявления. Как только сообщение записано, система уведомляет об этом автора исходного объявления посредством SMS. Получив такое сообщение, автор может позвонить на сервисный номер, указанный в сообщении и прослушать отставленное ему голосовое сообщение. При этом возможно ответить читателю посредством такого же голосового сообщения (с повторением SMS уведомления уже в другую сторону), так и прямо перезвонить читателю из голосового почтового ящика.

Смысл всего процесса состоит в том, что автор публикации может по своему усмотрению сохранять анонимность до тех пор, пока его не заинтересует какой-то из откликов. За счет использования SMS возможное общение может проходить без задержек. И при этом автор объявления может в любой момент времени остановить прием откликов, просто разорвав связь между кодом и своим номером (это также можно сделать, позвонив на сервисный номер). Это частая проблема для многих систем публикации. Как остановить отклики, если объявление уже утратило свою актуальность?

Естественно, что таким образом телефонное взаимодействие является скрытым для пользователей, но прозрачным для оператора (чтобы удовлетворять требованиям регуляторов).

Основой для такого рода сервисов служит Call Control API. Здесь и таится основная проблема. Вот, например, что предлагает в этой части GSM OneAPI [4].

OneAPI Call Control позволяет веб-приложениям управлять звонками. REST API использует HTTP команды POST, GET, PUT и DELETE. Ресурсы (объекты управления) представляются как URI. API предоставляет следующие возможности:

Организация вызова (звонка)
 Добавление участника к вызову
 Удаление участника вызова
 Получение информации о состоянии вызове
 Получение уведомлений о событиях во время звонка
 Завершение звонка
 Проигрывание аудио-файла
 Работа с тоновыми посылками

Например, следующий HTTP запрос осуществляет звонок [6]

```
POST /exampleAPI/thirdpartycall/callSessions HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
Accept: application/json
```

```
{"callSessionInformation": {
  "clientCorrelator": "104567",
  "participant": [
    {"participantAddress": "tel:+4912345678901",
     "participantName": "Max Muster"},
    {"participantAddress": "tel:+4412345678901",
     "participantName": "Peter E. Xample"}
  ]
}}
```

Отклик системы:

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/
thirdpartycall/callSessions/cs001
Content-Length: nnnn
Date: Mon, 28 Jun 2010 17:51:59 GMT
```

```
{"callSessionInformation": {
  "clientCorrelator": "104567",
  "participant": [
    {"participantAddress": "tel:+4912345678901",
     "participantName": "Max Muster",
     "participantStatus": "CallParticipantConnected",
     "resourceURL": "http://example.com/exampleAPI/1/
thirdpartycall/callSessions/cs001/participants/pt001",
     "startTime": "2010-06-28T17:50:51"},
    {"participantAddress": "tel:+4412345678901",
     "participantName": "Peter E. Xample",
     "participantStatus": "CallParticipantInitial",
     "resourceURL": "http://example.com/exampleAPI/1/
thirdpartycall/callSessions/cs001/participants/pt002"}
  ]
},
"resourceURL": "http://example.com/exampleAPI/1/
thirdpartycall/callSessions/cs001",
"terminated": "false"
}}
```

```
"terminated": "false"
}}
```

Заголовок Location содержит URI вызова. Именно к этому ресурсу будут отправляться все последующие запросы, касающиеся данного вызова. Например, так можно получить информацию о существующем звонке:

```
GET /exampleAPI/thirdpartycall/callSessions/cs001
HTTP/1.1
```

```
Host: example.com:80
```

Результат:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jun 2010 17:51:59 GMT
```

```
{"callSessionInformation": {
  "clientCorrelator": "104567",
  "participant": [
    {
      "participantAddress": "tel:+4912345678901",
      "participantName": "Max Muster",
      "participantStatus": "CallParticipantConnected",
      "resourceURL":
"http://example.com/exampleAPI/1/
thirdpartycall/callSessions/cs001/participants/pt001",
      "startTime": "2010-06-28T17:50:51"}
  ],
  {
    "participantAddress": "tel:+4412345678901",
    "participantName": "Peter E. Xample",
    "participantStatus": "CallParticipantInitial",
    "resourceURL":
"http://example.com/exampleAPI/1/
thirdpartycall/callSessions/cs001/participants/pt002"}
  ],
  "resourceURL": "http://example.com/exampleAPI/1/
thirdpartycall/callSessions/cs001",
  "terminated": "false"
}}
```

В целом, достаточно прозрачная и понятная в использовании система. При этом программировать сервисы можно как на стороне клиента (JavaScript), так и на сервере.

Но здесь и таится главная проблема – а как разработчику получить доступ к подобному API? OneAPI – не единственный подобный интерфейс, Parlay, например, предлагал нечто подобное [7]. Но вопрос именно в реализациях. Предложенные API совсем не спешат внедрять операторы связи, что, естественно закрывает путь для сервисов. В итоге, голосовые сервисы (если таковые вообще реализуются) оказываются привязанными к каким-либо специализированным платформам. Как минимум, они теряют переносимость.

Именно это и заставляет обратить взор на Voice Over IP

(VOIP) приложения.

III. VOIP КАК БАЗА ДЛЯ СЕРВИСОВ

Для телефонии, переход на VOIP – это отдельная большая задача (проблема). С этим можно ознакомиться, например, в статьях М.А. Шнепса [8,9]. С точки зрения разработчика же, VOIP является несомненным благом именно в силу наличия работающих программных интерфейсов для управления вызовами. Это еще один пример неповоротливости операторов связи, которая в итоге превратит их просто в канал для передачи данных.

Отсюда, в частности, интерес к Asterisk. Выбор этой платформы был обусловлен именно ее открытым API. Одна из первых работ была посвящена как раз реализации HTTP интерфейса для Call Control API [10, 11]. Следующая картина иллюстрирует спроектированный HTTP шлюз:

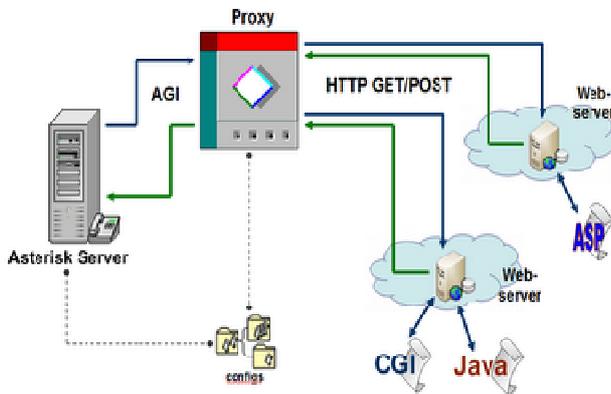


Рисунок 1. HTTP шлюз для Asterisk

Отметим, что по факту был спроектирован и реализован инструмент, подобный тому, что позднее был выпущен компанией Twilio [13].

Нам кажется, что это вполне интересное направление. По сути, речь идет о создании и использовании простого телекоммуникационного сервера приложений. Нечто подобное делает, например, Plivo [14]. Из других интересных задач в этой области можно назвать, например, автоматизацию программирования для подобного рода сервера [15]. Ряд интересных работ, связанных, в первую очередь, с администрированием систем был выполнен под руководством Д.Е. Гурьева.

Другие примеры приложений можно найти, например, в работе [16], посвященной как раз созданию мэшапов для средств массовой информации.

Следующая, на наш взгляд очень интересная тема – это использование собственно мобильного телефона в качестве такого сервера приложений. Удивительно, но подобного реализаций для, например, Android найти не удалось. Магистерская диссертация по этой теме [17] касалась уже устаревших моделей телефонов. Так что эта тема ждет своего исследователя. Тема управления звонками вообще довольно слабо представлена в современных мобильных ОС.

IV. ЗАКЛЮЧЕНИЕ

В работе рассматриваются вопросы проектирования и разработки телекоммуникационных сервисов. Речь идет о создании голосовых сервисов, непосредственно связанных с управлением звонками (вызовами). В работе описаны задачи, которые ранее рассматривались в рамках магистерских диссертаций и выпускных квалификационных работ.

БИБЛИОГРАФИЯ

- [1] Намиот Д., Сухомлин В. О проектах лаборатории ОИТ //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 5. – С. 18-21.
- [2] М.А. Шнепс-Шнеппе «Задачи производства изделий M2M: от простого к сложному» // Вестник связи, 2013, №9, 11-16.
- [3] 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.
- [4] Raivio, Y., Luukkainen, S., & Seppala, S. (2011, January). Towards Open Telco-Business models of API management providers. In System Sciences (HICSS), 2011 44th Hawaii International Conference on (pp. 1-11). IEEE.
- [5] D. Namiot and M. Snep-Sneppa "Context-aware data discovery", Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on, pp. 134-141, DOI: 10.1109/ICIN.2012.6376016.
- [6] GSMA Open API: <http://www.gsma.com/oneapi/> Retrieved: Feb, 2014
- [7] Намиот Д. Е., Жепич Д. Открытые интерфейсы Parlay X в узлах доступа NGN //Вестник связи. – 2006. – №. 1. – С. 34-37.
- [8] Шнепс М.А. От IN к IMS. О российской Системе-112: нерешенные задачи //International Journal of Open Information Technologies. – 2014. – Т. 2. – №. 1. – С. 1-11.
- [9] Шнепс М.А. От IN к IMS. О сетях связи военного назначения //International Journal of Open Information Technologies. – 2014. – Т. 2. – №. 2. – С. 1-11.
- [10] Schneps-Schneppa M., Namiot D. Machine-to-Machine Communications: the view from Russia //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 1. – С. 1-5.
- [11] Schneps-Schneppa, M., Maximenko, A., Namiot, D., & Malov, D. (2012, October). Wired Smart Home: energy metering, security, and emergency issues. In Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on (pp. 405-410). IEEE.
- [12] Schneps-Schneppa M., Namiot D., Ustinov A. A Telco Enabled Social Networking and Knowledge Sharing //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 6. – С. 1-4.
- [13] Stringer, Roger. Twilio Cookbook. Packt Publishing Ltd, 2013.
- [14] Di Fabrizio, Giuseppe, Thomas Okken, and Jay Wilpon. "Speech Mashups." Semantic Mashups. Springer Berlin Heidelberg, 2013. 205-235.
- [15] Bembieva, A., & Namiot, D. (2013). Voice Dialogs for Asterisk. International Journal of Open Information Technologies, 1(7), 24-27.
- [16] Шнепс-Шнеппе М., Намиот Д. Интеграция СМИ и телекоммуникаций //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 8. – С. 7-12.
- [17] Асиненко А., Намиот Д. Программная обработка голосовых вызовов на мобильных телефонах //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 6. – С. 18-21.

On telecom services

Gouriev D.E., Namiot D.E., Sneps M.A.

Abstract—This article examines the development of telecommunication services and describes the experience of their creation and implementation. Discussed services were the basis of many qualification works (Master's theses, diplomas) performed in the Open Information Technologies Lab Faculty of Computational Mathematics and Cybernetics of MSU. The article analyzes the obtained results and discusses the future directions for research.

Keywords— telecom, services, voice services, mobile services.