

УДК 32.95.

**Ормонова Ирсалат Абдырахмановна**

доцент,

Ошского технологического университета им. М.М.Адышева

**Ormonova Irsalat Abdyrakhmanovna**

доцент,

М.М.Адышев ат. Ош технологиялык университети

**Ormonova Irsalat Abdyrakhmanovna**

assistant professor,

Osh Technological University named after. M.M.Adysheva

**Ормонов Мурза**

инженер программист,

Ошского технологического университета им. М.М.Адышева

**Ormonov Murza**

инженер программист,

М.М.Адышев ат. Ош технологиялык университети

**Ormonov Murza**

programmer engineer,

Osh Technological University named after. M.M.Adysheva

**ИССЛЕДОВАНИЯ О ТЕЛЕКОММУНИКАЦИОННЫХ СЕРВИСАХ**

**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 Ош ТУ в лаборатории Кибернетика и информационных технологий факультета (КиИТ) Ош ТУ. В статье анализируются полученные результаты, приводятся примеры квалификационных работ и перспективные направления для исследования.

**Ключевые слова:** телеком, сервисы, голосовые услуги, мобильные сервисы.

**ТЕЛЕКОММУНИКАЦИЯ КЫЗМАТТАРЫН ИЗИЛДӨӨ**

**Аннотация.** Макалада телекоммуникация кызматтарын өнүктүрүү маселелери аларды түзүү жана ишке ашыруу тажрыйбасы каралат. Ош ТУнун бүтүрүүчүлөрү Кибернетика жана маалымат технологиялары (КЖИТ) бүтүргөн көптөгөн квалификациялык иштердин (магистрдик диссертациялардын, дипломдордун) темаларын аныктоого негиз болгон. Ош ТУнун лабораториясын кызматтын ушул түрүн иштеп чыгууга жана иштетүүгө маселелер байланышкан. Макалада алынган квалификациялуу иштердин мисалдары жана изилдөө үчүн келечектүү багыттар келтирилген натыйжалар каралат.

**Негизги сөздөр:** телекоммуникация, кызматтар, үн кызматтары, мобилдик кызматтар.

**RESEARCH ON TELECOMMUNICATION SERVICES**

**Annotation.** The article discusses the issues of developing telecommunication services, analyzes the experience of their creation and implementation. Issues related to the devel-

development and operation of this kind of services were the basis for determining the topics of many qualifying works (master's theses, diplomas) completed by graduates of Osh TU in the laboratory of Cybernetics and Information Technologies of the Faculty (KiIT) of Osh TU. The article analyzes the results obtained, provides examples of qualifying work and promising areas for research.

**Key words:** telecom, services, voice services, mobile services.

## **ВВЕДЕНИЕ**

**В данной работе рассматриваются** вопросы, связанные с проектированием, разработкой и эксплуатацией телекоммуникационных услуг. Мотивом для написания данной работы в определенной степени послужила диссертация [1].

В работе изложены результаты и тематика магистерской диссертации по кибернетике и информационным технологиям (КИТ) факультета КИТ Ошского Технологического Университета им. М. Адышева.

Можно отметить, что большая часть предыдущих исследований была посвящена телекоммуникационным услугам. Поэтому данная статья является попыткой систематизировать эти работы.

Не претендуя перед собой задачу охватить всю картину в рамках одной статьи, следует отметить, что развитие сервисов в отечественных изданиях и учебных курсах в высшей школе в некоторой степени осталось за рамками рассмотрения.

Точнее, они практически не принимаются во внимание. Традиционно академические издания, связанные с телекоммуникациями, как правило, охватывают только описание протоколов взаимодействия.

В терминах семи уровневой модели сети они относятся к нижним уровням. Создается впечатление, что все усилия направлены на разработку телекоммуникационного оборудования, но это редко встречается в национальных версиях [2].

Таким образом, вся область остается в не рассмотренной. Например, выпускники вряд ли будут знать о таких актуальных проектах, как Интернет вещей [3] или европейский проект FI-WARE.

Это происходит не только потому, что статья была написана для академического журнала, издаваемого на кафедре, связанной с математикой или информатикой, а не с телекоммуникациями.

Например, анализ всех последних проектов связанных выше упомянутыми направлениями IoT, оказывает, что они в основном связаны с программированием.

Конечным выражением и видимой частью всего являются именно сервисы. Именно программы их реализуют. Анализ и разработка протокола как раз связаны с разработкой аппаратного обеспечения, а системная архитектура - это, например, облачные технологии, распределенные системы и обработка больших данных.

В этой статье особое внимание уделяется голосовым услугам. Традиционная часть телефонии - это звонок. Именно с этого все начиналось. И все продолжается. Не важно, как осуществляется звонок - по традиционному каналу или по VoIP

## **Идеология сервисов**

Основная идея создания голосовых сервисов заключается в том, что звонки как-то образом встраиваются (embedded) в другие приложения.

Собственно говоря, это относится и к другим коммуникационным сервисам.

Нет(и не будет) «золотой» услуги(приложения), которая бы одним махом решала все проблемы абонентов и приносила операторам постоянный доход. Еще один важный момент - признать, что введение платных услуг(услуг за которые взимается дополнительная плата) не является услугой. Оно также не решает проблем ни пользователя, ни оператора.

Время премиальных услуг (если таковые вообще существовали) закончилась. Звонки-это базовая услуга. Премиальные здесь просто не используются(или не используются в коммерчески значимых объемах).

В тоже время операторы связи вполне успешно берут деньги за базовые услуги. А мы считаем, что цель услуг-помогать. Бизнес-модель услуг должна быть направлена на увеличение потребления базовых услуг. Какие услуги операторы умеют и успешно собирают деньги за них?

Интеграция обработки вызовов в сторонние приложения приводит к тому, что звонок (голосовая связь) становится еще одним каналом взаимодействия с приложением. И в тоже время такое взаимодействие естественным образом увеличивает потребление базового коммуникационного сервиса.

Вызов, встроенный в приложение, подразумевает, конечно же, программную обработку этого вызова. Что означает эта обработка и какие функции она включает? Программная обработка-это всегда возможность выполнить свой собственный код (функцию, процедуру, модуль ит. д.)при поступлении вызова на определенный номер. Технически требуется совсем немного функций, а их сочетание дает возможность строить сервисы. Таких базовых функций можно насчитать шесть.

Во-первых, необходимо совершить звонок .Ваш собственный программный код должен уметь звонить на заданный номер и выполнять какие-то действия по факту входящего звонка или отсутствия ответа.

Во-вторых, вызов должен быть завершен. Это означает, что вы должны выполнить собственный код по факту входящего вызова, прочитать (определить) номер вызываемого абонента с помощью этого кода и завершить вызов.

Примером использования может служить любая служба телефонного голосования. Код, выполняемый по факту входящего звонка, например, просто увеличивает счетчик в некоторой базе данных. Затем требуется ввести номер звонящего, чтобы предотвратить повторное голосование. С практической точки зрения именно это и есть завершение звонка.

Программа-обработчик обновляет базу данных со статистикой голосования и просто завершает вызов(«вешает трубку»).

Следующая функция-воспроизведение звукового файла. Типичный пример-автоответчик. Программа, принимающая звонок (т.е.» вешающая трубку»),решает, какой звуковой файл будет уместен .Для этого программа может руководствоваться, например, номером звонящего, временем суток, погодой и другими доступными данными. В качестве контекста может выступать ,например, контекстно-ориентированное приложение[5]

Затем выбранный аудио файл проигрывается(«прослушивается»)звонящему. Аналогичным образом эта функция должна работать при совершении телефонного звонка. Сделайте звонок, дождитесь ответа и воспроизведите аудиофайл.

Следующий момент-пере адресация вызовов. Входящие звонки переадресуются на другой номер. Программа принимает звонок, определяет новый номер по собственному алгоритму и пере адресует абонента на новый номер.

#### **Примеры услуг-Виртуальный номер.**

Следующая функция-запись аудио файлов. Программа принимает звонок («снимает трубку») и записывает все, что сказал звонящий (или программа на вызове). Пример услуги-голосовая почта. Последняя функция - взаимодействие с пользователем. Прием тональных посылок. То, что вызываемый абонент (другая программа) указывает с помощью дополнительного набора.

Рассмотрим пример услуги (реализованной на практике на платформе Audiotele). В системе передачи (классификации) рекламы пользователь (создатель рекламы) указывает код для связи в качестве контактной информации. Этот код привязан к номеру мобильного телефона, который можно получить, зарегистрировав мобильный телефон на веб-портале проекта или позвонив на сервисный номер.

#### **Примеры услуг-Виртуальный номер.**

Следующая функция-запись аудио файлов. Программа принимает звонок («снимает трубку») и записывает все, что сказал звонящий (или программа на вызове). Пример услуги-голосовая почта. Последняя функция - взаимодействие с пользователем. Прием тональных посылок. То, что вызываемый абонент (другая программа) указывает с помощью дополнительного набора.

Рассмотрим пример услуги (реализованной на практике на платформе Audiotele ).

В системе передачи (классификации) рекламы пользователь (создатель рекламы) указывает код для связи в качестве контактной информации. Этот код привязан к номеру мобильного телефона, который можно получить, зарегистрировав мобильный телефон на веб-портале проекта или позвонив на сервисный номер.

При размещении рекламы автор указывает один код для связи.

Если читателю рекламы необходимо связаться с автором (его заинтересовало авторское объявление) . Этот код не нужно запоминать ,его можно найти непосредственно в рекламе.

Указав код, читатель может записать голосовое сообщение автору рекламы. Как только сообщение будет записано, система уведомит автора оригинальной рекламы по SMS. Рекламодатель, получивший такое сообщение ,может позвонить по указанному в сообщении сервисному номеру и прослушать присланное голосовое сообщение.

Это же голосовое сообщение можно использовать для ответа читателю (SMS-уведомление повторяется в обратном направлении )или для звонка читателю непосредственно с голосового почтового ящика.

Смысл всего этого процесса в том, что автор публикации может по своему усмотрению оставаться анонимным до тех пор, пока его не заинтересует один из ответов;

Использование SMS позволяет осуществлять связь без задержек .В тоже время рекламодатель может в любой момент прекратить получать ответы, просто отключив код от своего номера(это также можно сделать, позвонив на сервисный номер). Это общая проблема многих издательских систем. Как остановить отклик, если объявление уже

Естественно ,при таком способе телефонное взаимодействие не заметно для пользователя, но прозрачно для оператора (для соблюдения нормативны требований).

API Call Control является основой для такого сервиса. Именно здесь и кроются основные проблемы. Например, вот что предлагает GSMA One API [4]в этой части

One Call Control API позволяет веб-приложениям управлять вызовами; REST API использует HTTP-команды POST, GET, PUT и DELETE. Ресурсы (объекты управления) представлены в виде URI; API обеспечивает следующие функции:

- Организация звонков(звонок) Добавление участников в звонок Удаление участников из звонка

- Получение информации о состоянии вызова Получение уведомлений о событиях во время вызова Завершение вызова

- Воспроизведение аудио файлов Работа с тональными посылками.

- Например, следующий HTTP-запрос инициирует вызов

```

":{ "clientCorrelator":
"103467", "participant terminated": "false};
[{"participant Address": "tel:+996505281290", "participant Name":
" Aigyl Binazarova", "participant Status": "Call Participant Connected", "resource URL":
"http://example.com/exampleAPI/1/
Thir dparty call/call Sessions/cs001/participants/pt001", "start Time": "2023-06-28
T17:50:51"},
{"participant Address": "tel:+ 996772775899 ",
"participant Name "Nazgyl Ormonkylova", "participant Status":
"Call Participant Initial", "resource URL": "http://example.com/exampleAPI/1/thirdparty-
call/call Sessions/cs001/participants/pt002"},
"resource URL": "http://example.com/exampleAPI/1/"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/example API/thir dparty call /call Sessions/cs001 HTTP/1.1
```

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

### Результат:

```
HTTP/1.1 200 OKContent-Type: application/json Content-Length: nnnn
```

```
Date: Mon, 28 Jun 2023 17:51:59 GMT
```

```
{"call Session Information": {"client Correlator": "103467", participant":996505281290}}
```

```
[{"participant Address": "tel:+996505281290", "participant Name": " Aigyl Binazarova ";
"participant Status": "Call Participant Connected", "resource URL":
```

```
"http://example.com/exampleAPI/1/ thir dpartycall/ call Sessions/cs001/participants/
pt001",
```

```
"start Time": "2023-06-28T17:50:51"},"participant Address": "tel:+ 996772775899",
"participant Name": " Nazgyl Ormonkylova "; "participant Status": "Call Participant Initial",
"resource URL":
```

```
"http://example.com/exampleAPI/1/ thir dparty call/call Sessions/cs001/participants/
pt002"}},
```

```
"resource URL": "http://example.com/exampleAPI/1/ thir dparty call/call Sessions/
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.

### VOIP как база для сервисов

Для телефонии переход на VOIP является серьезным вызовом.

С точки зрения разработчика, VOIP-несомненное благо именно из-за наличия практичных программных интерфейсов для управления вызовами. Это еще один пример неповоротливости операторов связи, которая в конечном итоге превратит их в простые каналы для передачи данных.

Поэтому особый интерес вызвала платформа Asterisk. Платформа была выбрана именно из-за ее открытого API. Одна из первых задач была посвящена реализации HTTP-интерфейса API управления вызовами [10, 11]. На следующей схеме показан разработанный HTTP-шлюз:

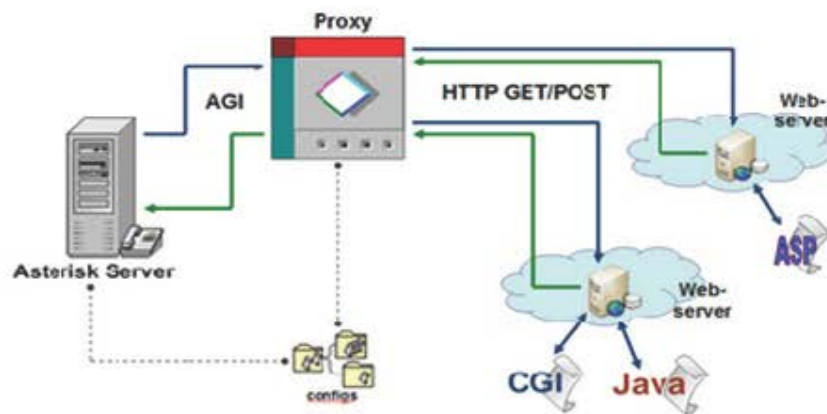


Рис. 1. HTTP шлюз для Asterisk

Отметим, что на практике был разработан и реализован инструмент, аналогичный тому, который позже выпустила компания Twilio.

Это кажется очень интересным направлением. По сути, создается и используется простой сервер коммуникационных приложений. Нечто подобное делает, на пример, компания Plivo. Другие интересные задачи в этой области включают, например, автоматизацию программирования такого типа серверов. Другие области применения включают, например, задачу создания mash-ups для медиа.

Следующая очень интересная тема, которую мы рассматриваем - это использование самого мобильного телефона в качестве сервера приложений. Удивительно, но мы не нашли ни одной реализации, подобной, например, Android. Магистерская диссертация на эту тему была посвящена уже устаревшей модели. Поэтому данная тема еще ждет исследователей. В целом тема управления вызовами встречается в современных мобильных операционных системах реже.

## 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.

## Литература;

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. Sneps-Sneppe “Context-aware data discovery”, Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on, pp. 134-141, DOI: 10.1109/ICIN.2012.6376016. GSMA Open API: <http://www.gsma.com/oneapi/> Retrieved: Feb, 2014.
6. Намиот Д. Е., Жепич Д. Открытые интерфейсы Parlay X в узлах доступа NGN // Вестник связи. – 2006. – №. 1. – С. 34-37.
7. Сопубеков Н.А. Использование информационных систем в банковской деятельности [Текст]: / Н.А. Сопубеков, А.М. Маткалыков // Известия Ошского технологического университета. –Ош, 2017, №4. –С. 32-36.
8. Сопубеков Н.А. Выбор метода планирования сети доступа NGN. [Текст]: / Н.А. Сопубеков, Н.Б. Назарбеков, А.М. Карабаева // Известия Ошского технологического университета. –Ош, 2018. №3. –С. 90-96.
9. Schneps-Schneppe M., Namiot D. Machine-to-Machine Communications: the view from Russia //International Journal of Open Information Technologies. – 2013. – Т. 1. №. 1.