

**РАЗРАБОТКА УСЛУГ С ДОБАВОЧНОЙ СТОИМОСТЬЮ
С ИСПОЛЬЗОВАНИЕМ АРХИТЕКТУРЫ PARLAY**

**DEVELOPMENT OF VALUE ADDED SERVICES
USING PARLAY ARCHITECTURE**

Аннотация. Проведены исследования архитектуры и интерфейсов, а также анализ функциональной модели последней версии Parlay–шлюза. При анализе функциональной модели собрана воедино вся информация, касающаяся спецификации интерфейсов Parlay, в соответствии с последними документами Европейского института стандартизации телекоммуникаций ETSI, получившими статус стандарта. Раскрываются функциональные возможности элементов Parlay–шлюза, исследуются процедуры управления предоставлением услуг. Разработана модель услуги, реализующей прием заявки от абонента на получение контента и посылку пользователю заказанного контента в виде SMS–сообщений с использованием интерфейсов Parlay–шлюза.

Summary. The researches of architecture and interfaces, and also analysis of functional model of the last version of Parlay gateway are made. All information from the Parlay specifications is collected together for the analysis of functional model in accordance with the last ETSI documents. Parlay gateway elements functional possibilities and the services management procedures are explored. The model of the service is developed. It can receive the subscriber's request on sending to and receiving the requested content by the user by SMS using the Parlay interfaces.

В настоящее время проблема, связанная с отсутствием в украинских операторов телекоммуникаций опыта разработки и внедрения услуг с добавочной стоимостью – VAS (Value Added Services), приводит к малому выбору на отечественном рынке таких услуг и невысокому спросу на них среди украинских абонентов.

К VAS услугам относятся такие, например, как интеллектуальные услуги, мультимедийные услуги, услуги определения местоположения абонента LBS (Location Based Services) и инфокоммуникационные услуги. Широкому внедрению таких дополнительных видов обслуживания абонентов в зарубежных странах способствуют следующие причины:

- рынок телефонной связи за рубежом находится в состоянии насыщения, что стало одной из причин устойчивого снижения среднего ежемесячного дохода оператора с одного пользователя;
- либерализация мирового рынка услуг связи, сделав его более доступным для новых операторов и поставщиков услуг, обострила конкуренцию;
- у абонентов (прежде всего корпоративного сектора) появилась потребность в услугах связи, которые можно было бы гибко настраивать в соответствии с их требованиями, например, маршрутизировать вызовы на разные телефонные номера в зависимости от дня недели или времени суток, обеспечивать мобильность терминала и абонента.

Данные обстоятельства, наряду со стремлением операторов к развитию телекоммуникационной сети, борьбой с несанкционированным доступом и задолженностями абонентов вынуждают операторов внедрять VAS услуги, чтобы удержать имеющихся абонентов и найти новых потребителей своих услуг.

Для работы в этом направлении в 1998 г. создан Консорциум Parlay Group, который к настоящему времени включает в себя более 60 телекоммуникационных операторов и других компаний [1, 2]. Первоначальной целью Консорциума была организация взаимодействия сетей связи при предоставлении услуг, для чего были разработаны открытые интерфейсы прикладного программирования API (Application Programming Interface) для архитектуры открытого доступа к услугам OSA (Open Service Access).

Однако для широкого внедрения VAS услуг операторами телекоммуникаций необходимо решить еще ряд задач, среди которых следует выделить разработку моделей VAS услуг с использованием интерфейсов Parlay и последующую программную реализацию услуг на объектно–ориентированных языках программирования.

Целью настоящей работы является анализ архитектуры и интерфейсов Parlay для определения целесообразности их применения в Украине и разработка на основе выполненного анализа модели

услуги, реализующей прием заявки от абонента на получение контента и посылку пользователю заказанного контента в виде служебных SMS-сообщений.

1. Анализ архитектуры и интерфейсов Parlay. Анализ архитектуры предоставления VAS услуг следует проводить по ряду таких признаков:

- организация взаимодействия сетей связи при предоставлении услуг;
- организация средств аутентификации, авторизации и системы расчетов;
- качество услуг и меры его гарантированного обеспечения;
- методы тарификации и взаиморасчетов между операторами, между операторами и пользователями по услугам;
- особенности и тенденции развития услуг на базе новых технологий.

В архитектуре OSA/Parlay между телекоммуникационными сетями и сервером приложений устанавливается OSA/Parlay-шлюз (рис. 1), разделяющий уровни сервисной и базовой сетей [1, 3]. Базовая сеть в свою очередь включает уровень управления, транспортный уровень и сеть доступа. Каждый из сетевых элементов выходит на шлюз (Gateway) по своему протоколу, а задача шлюза состоит в том, чтобы свести все протоколы к единым интерфейсам API. Тогда приложения можно создавать без учета особенностей нижележащих сетей и следует только строго придерживаться интерфейсов API.

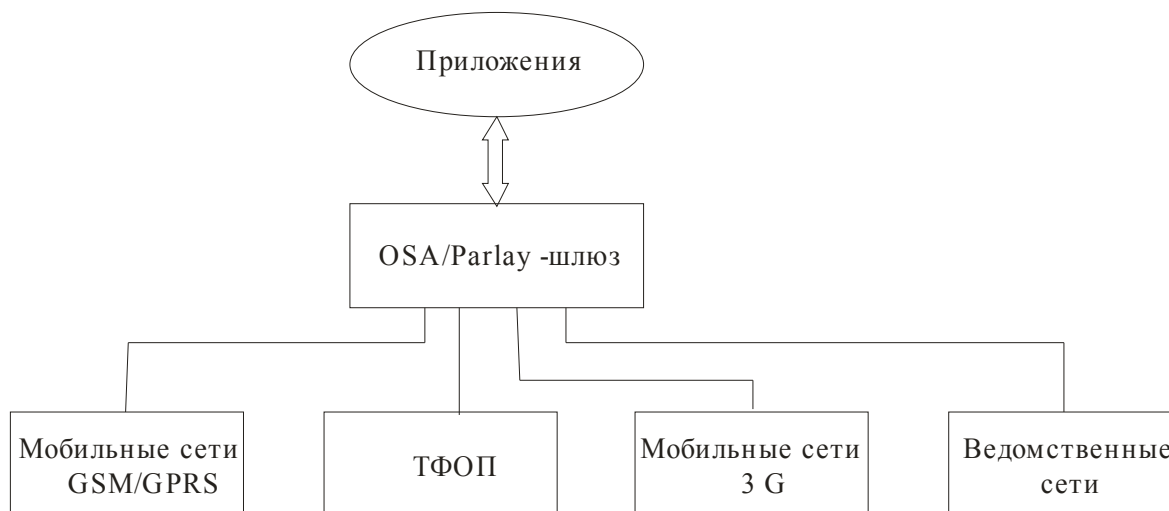


Рисунок 1 – Положение Parlay-шлюза в архитектуре OSA/Parlay

Вышеупомянутый шлюз, отвечающий за реализацию API, носит название сервера доступа к услугам – SCS (Service Capability Server).

Приложения, устанавливаемые на серверах приложений, используют функциональность серверов доступа к услугам SCS, определенную в OSA/Parlay [1]. Эта функциональность реализуется посредством API: в SCS реализована серверная часть API, а в приложениях — клиентская. Серверы доступа к услугам взаимодействуют с такими элементами базовой сети, как коммутаторы, регистры местоположения, интерактивные системы голосовой почты, системы мобильного позиционирования, шлюзовые узлы поддержки услуг GPRS (General Packet Radio Service), биллинговые системы и т. д. Серверы доступа к услугам являются логическими элементами, и в спецификациях не фиксируется их физическая реализация в сети. В принципе SCS может быть реализован как автономный сетевой узел, либо в существующих узлах базовой сети.

Открытые интерфейсы Parlay являются объектно-ориентированными программами, написанными на языке Java, выполняются виртуальной машиной JVM (Java Virtual Machine) и состоят из совокупности классов (рис. 2) [1, 3, 5]. Основными являются две группы интерфейсов: Parlay-остов (Framework, Fw) и Parlay-услуги (Parlay Service), которые также называют сервисными компонентами.

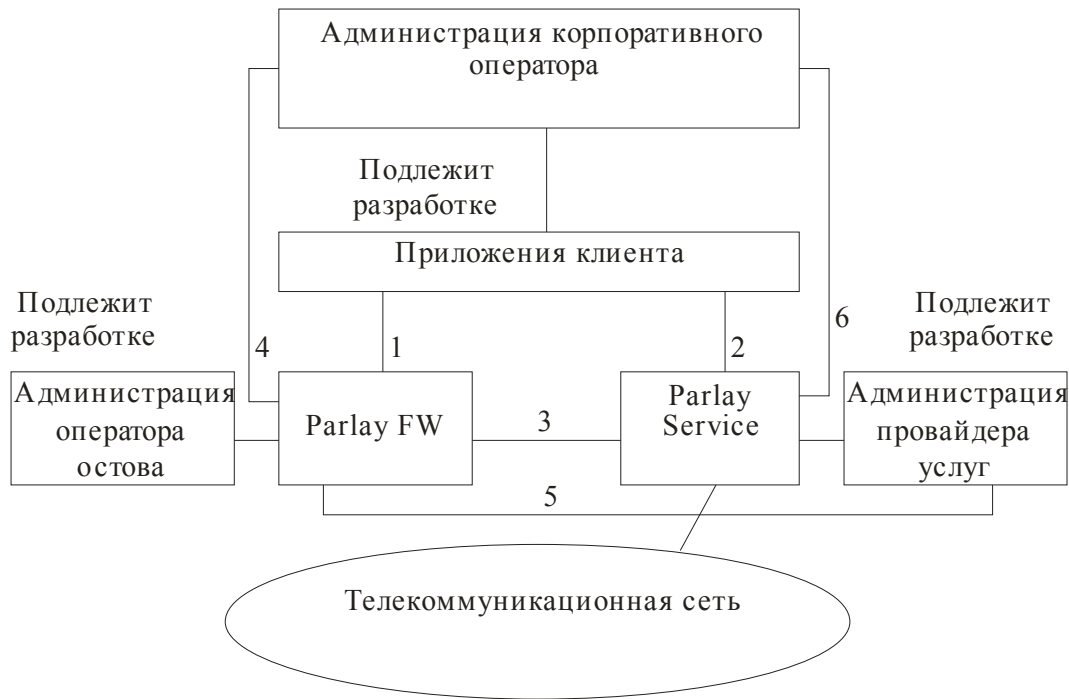


Рисунок 2 – Классы и интерфейсы Parlay

Спецификации интерфейсов Parlay (версия 4.1) опубликованы, в частности, в виде рекомендаций ETSI (ES 201 915–№ тома (версия 1.2.1)) и содержат 14 томов [4]:

Том 1: Обзор.

Том 2: Определения общих данных.

Том 3: Остов.

Том 4: Управление обслуживанием вызова:

Подраздел 1: Общие сведения по управлению обслуживанием вызова.

Подраздел 2: Общее управление обслуживанием вызова.

Подраздел 3: Управление обслуживанием многостороннего вызова.

Подраздел 4: Управление обслуживанием мультимедийного вызова.

Подраздел 5: Управление обслуживанием конференц-вызова.

Том 5: SCF Взаимодействие с пользователем.

Том 6: SCF Мобильность.

Том 7: SCF оконечных устройств, свойства терминала.

Том 8: SCF Управление сеансом передачи данных.

Том 9: SCF Службы сообщений.

Том 10: SCF Менеджер соединения (Управление качеством QoS).

Том 11: SCF Управление счетами.

Том 12: SCF Тарифы (тарификация контента).

Том 13: SCF Управление правилами/политикой.

Том 14: SCF Управление сервисом Присутствие и Доступность.

Интерфейс приложения – Framework (интерфейс 1, рис. 2) устанавливает требования к средствам аутентификации, авторизации и системам расчетов и определяет нормы на качество услуг и меры его гарантированного обеспечения. Данный интерфейс обеспечивает механизмы [3]:

- аутентификации приложений;
- авторизации приложений;
- инвентаризации (discovery) Framework и функций сервисных компонентов (приложение получает доступ к интерфейсам Framework и информацию о доступных функциях сервисных компонентов);
- установления сервисного соглашения;
- доступа к функциям сервисных компонентов.

Обратим внимание на меры гарантированного обеспечения качества услуг, описанные в томе 10 спецификаций OSA/Parlay. В ходе получения доступа приложения к ресурсам базовой сети

происходит подписание сервисного соглашения SLA (Service Level Agreement), в котором оговариваются точные параметры обслуживания, такие как: потери пакетов, задержки, джиттер и действия при избыточном трафике. Также в концепции OSA/Parlay предусмотрены готовые шаблоны уровней обслуживания, с заранее заданными значениями параметров.

Интерфейс Framework – сервисные компоненты (интерфейс 3, рис. 2) обеспечивает механизм регистрации функций сервисных компонентов в Framework.

Интерфейс Framework – оператор услуг (интерфейс 4, рис. 2) обеспечивает механизм подписки контрактного соглашения (Сервисного Контракта) между оператором услуг и Framework.

В концепции Parlay подробно рассматривается бизнес–модель подписки приложений на ресурсы сервисных компонентов [3]. Любые ресурсы не бесплатны, и не могут быть предоставлены сервисному приложению в неограниченном количестве. Подписка на ресурсы осуществляется оператором услуг (Enterprise Operator) через Framework. Таким образом, данный интерфейс определяет методы тарификации и взаиморасчетов между операторами, между операторами и пользователями по услугам (см. рис. 2).

Здесь оператор услуг выступает в роли заказчика ресурсов сервисных компонентов, сервисное (клиентское) приложение – в роли их потребителя, а Framework – в роли розничного продавца ресурсов. Все они представлены в домене Framework программно в качестве соответствующих объектов со своими свойствами.

Механизм подписки сервисного контракта заключается в установлении на определенных условиях ассоциированных связей между сервисными приложениями, принадлежащими данному оператору услуг, и необходимыми ему функциями сервисных компонентов. Таким образом, технология OSA/Parlay реализует концепцию «бизнес для бизнеса» (B2B) [3, 6].

Оператором услуг может быть сервис–провайдер, или любой другой участник рынка, который выступает в его роли (оператор телекоммуникаций).

На рис. 2 показаны также направления будущих разработок в рамках концепции Parlay, а именно, интерфейсы административного управления:

- между клиентом и администрацией корпоративного оператора;
- между Parlay–остовом и администрацией остова;
- между Parlay–услугами и администрацией провайдера услуг.

Отдельное внимание Консорциумом Parlay Group уделяется вопросу особенностей и тенденций развития услуг на базе новых технологий, в частности Web–технологий. Поэтому разработана концепция Parlay X, представляющая из себя фактически концепцию Web–услуг, применимую к телекоммуникационной области [3, 5, 7].

Основная цель заключается в выборе достаточно мощного, но простого набора функций телекоммуникационных сетей [3]:

- каждый модуль услуг (building block) представляет абстракцию некоторого набора интерфейсов Parlay;
- возможности каждого модуля могут быть однородны или неоднородны;
- взаимодействие между приложением и модулем услуг (Parlay X API) должно быть реализовано посредством обмена XML–сообщениями;
- модули не должны содержать специфику приложений, т.е. логика сервиса должна полностью определяться разработчиком.

В таком случае в разработке приложений Parlay X смогут принять участие web–разработчики, специализирующиеся на языке XML (eXtensible Markup Language).

Приложения Parlay X могут взаимодействовать с телекоммуникационными сетями напрямую или через классический Parlay–шлюз. В последнем случае платформа Parlay X Web Services, которая и реализует Web–услуги Parlay X, поддерживает существующие открытые прикладные программные интерфейсы Parlay APIs в сторону Parlay шлюза.

Для разработки приложений Parlay X основным средством описания логики телеком–услуги является скрипт XML. В качестве транспортных средств чаще всего используются:

- CORBA (Common Object Request Broker Architecture) – универсальный объектно–ориентированный механизм взаимодействия приложений в распределенных системах, стандартизованный организацией OMG [3];
- SOAP (Simple Object Access Protocol) – упрощенный протокол общения объектов в распределенных системах, специфицирован организацией W3C (World Wide Web Consortium) [3].

Концепція Parlay X знаходиться в процесі розвитку. На даному етапі не вирішені наступні задачі концепції Parlay X:

- вибір параметрів використовуваного додатком сервіса;
- аутентифікація додатку;
- авторизація додатку і ряд інших.

Із сказаного вище випливає, що, як архітектури відкритого доступу до послуг, концепції Parlay і Parlay X пропонують рішення проблем, перерахованих в початку статті, що є очевидним їх достоїнством.

С точки зору оточення для створення послуг, концепції Parlay і Parlay X мають наступними достоїнствами [8]:

- невелике час від виникнення ідеї деякої послуги до її постачання кінцевим користувачам;
- підтримка гнучкої моделі оплати послуг;
- витрати діяльності мінімізовані для кожної послуги;
- підтримка взаємодії з діловими партнерами;
- підтримка стратегії маркетингу, орієнтованої на кінцевого користувача;
- розвиток послуг незалежно від постачальників мережевого обладнання.

Поэтому Консорциум 3GPP определил сервер OSA SCS как одну из трех категорий окружения для создания услуг для платформы IMS (IP Multimedia Subsystem), предусмотрев на прикладном уровне установку Parlay-шлюза, обеспечивающего открытый интерфейс для предоставления услуг третьими сторонами.

Следует также подчеркнуть, что в национальных стандартах Украины в настоящее время отсутствует упоминание концепций Parlay и Parlay X, вероятно, по причине медленного развития стандартов и отсутствия законодательной базы, определяющей правовые и технико-экономические условия внедрения новых услуг по доступным ценам.

2. Розробка моделі послуги з використанням інтерфейсів Parlay. Розробимо модель послуги, що надсилає користувачеві заздалегідь замовлений контент у вигляді SMS-повідомлень, з використанням універсального мови моделювання UML, так як створені з допомогою UML моделі можуть бути реалізовані на різних мовах програмування.

Посылка пользователю заранее заказанного контента в виде SMS-сообщений обеспечивается двумя сервисными компонентами шлюза Parlay: Call Control SCF (управление обслуживанием вызова) и User Interaction SCF (взаимодействие с пользователем).

Для рассмотрения принципа работы модели приведем некоторые сведения из спецификаций Parlay. В терминах Parlay системные компоненты называются услугами, а сами VAS услуги именуются приложениями. Далее будем следовать данной терминологии.

Чтобы использовать одну из услуг шлюза Parlay, необходим менеджер услуги. Например, услуга “Взаимодействие с Пользователем” имеет менеджера типа IpUIManager.

Каждому менеджеру услуги нужно иметь callback (интерфейс обратной связи) менеджера услуги, в данном случае – IpAppUIManager. Другими словами, каждому объекту, используемому приложением, необходим callback, который шлюз сможет использовать, чтобы взаимодействовать с приложением. Callback-классы можно отличить по тому признаку, что их имена начинаются с IpApp вместо Ip.

Менеджер услуги содержит методы, которые приложение может использовать. IpUIManager, например, содержит метод sendInfoReq(), который может использоваться, чтобы послать сообщение. Имя метода заканчивается на Req, что указывает на то, что метод асинхронный.

Как только шлюз послал сообщение, он извещает приложение с помощью другого метода, находящегося в классе callback. Этот метод можно отличить по тому, что его имя заканчивается на Res. В данном случае результат будет получен с помощью sendInfoRes, который расположен в классе IpAppUIManager.

Почти все методы OSA/Parlay асинхронные. Синхронные методы можно отличить по тому признаку, что их имя не заканчивается на Req. Синхронный метод непосредственно возвращает результат.

Работа рассматриваемой услуги начинается с приема заявки от абонента. Этот этап реализован в модели в виде интерактивного опроса абонента, для этого используем такие интерфейсы и методы (рис. 3):

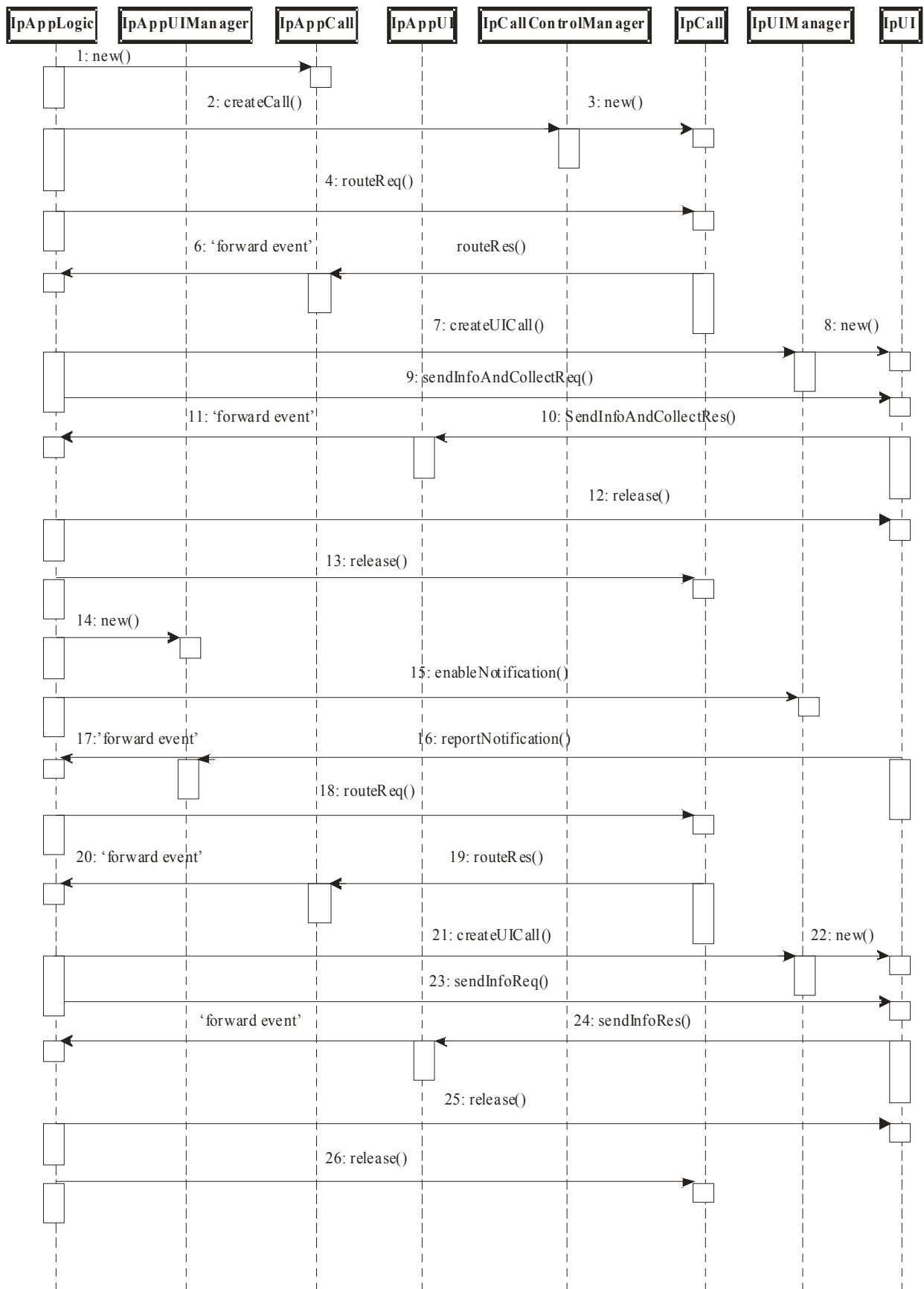


Рисунок 3 – Поточкова діаграма інтелектуальної послуги надання контенту за допомогою SMS-повідомлень

- 1) Приложение вызывает метод new() для того, чтобы создать объект, реализующий интерфейс IpAppCall.
 - 2) Методом createCall() приложение делает запрос объекту, реализующему интерфейс IpCallControlManager, создать объект, реализующий интерфейс IpCall (вызов).
 - 3) Объект, реализующий интерфейс IpCallControlManager, вызывает метод new(), создавая объект, который реализует интерфейс IpCall.
 - 4) Приложение вызывает метод routeReq() интерфейса IpCall, который выполняет маршрутизацию вызова к указанному адресу.
 - 5) Сервисный интерфейс вызовом метода routeRes() возвращает результат обслуживания вызова на свой интерфейс обратной связи.
 - 6) Такое сообщение используется прикладным интерфейсом IpAppCall обратной связи для передачи предыдущего сообщения интерфейса приложения.
 - 7) Приложение посылает запрос createUICall() объекту, реализующему интерфейс IpUIManager, который управляет вызовом для взаимодействия с абонентом, создать объект, реализующий интерфейс IpUI (взаимодействие с абонентом).
 - 8) Допустим, что характеристики такого объекта (взаимодействия с абонентом) отвечают критериям, посланным оператором связи, тогда объект, реализующий интерфейс IpUIManager, вызывает метод new(), создавая объект, который реализует интерфейс IpUI.
 - 9) Приложение вызывает метод sendInfoAndCollectReq() интерфейса IpUI, который посылает по указанному адресу справочную информацию и одновременно принимает ответ абонента. Данный метод позволяет это выполнить, послав абоненту как SMS-сообщение, так и голосовую справку. Сессия таких запросов позволяет абоненту задать такие параметры услуги, как адрес ресурса, тематика контента, время поступления сообщения и др.
 - 10) Сервисный интерфейс вызовом метода sendInfoAndCollectRes() возвращает результат взаимодействия с абонентом на свой интерфейс обратной связи.
 - 11) Такое сообщение используется прикладным интерфейсом IpAppUI обратной связи для передачи предыдущего сообщения интерфейсу приложения.
 - 12) Приложение вызывает метод release() интерфейса IpUI для освобождения UI объекта.
 - 13) Приложение вызывает метод release() интерфейса IpCall для освобождения Call объекта.
- На следующем этапе модель услуги ожидает поступления необходимого контента, т.е. поступления специального отчета, который становится сигналом к обработке контента и отправке его абоненту. Для этого использованы такие интерфейсы и методы (см. рис. 3).
- 14) Приложение вызывает метод new() для того, чтобы создать объект, реализующий интерфейс IpAppUIManager.
 - 15) Методом enableNotification() интерфейса IpUIManager приложение сообщает, что может принимать из сети специальные отчеты.
 - 16) При возникновении в сети события метод reportNotification() передает специальный отчет на интерфейс обратной связи IpAppUIManager.
 - 17) Такое сообщение используется прикладным интерфейсом IpAppUIManager обратной связи для передачи предыдущего сообщения интерфейсу приложения.
 - 18) – 22) Повторяются шаги со 2) по 8).
 - 23) Приложение вызывает метод sendInfoReq() интерфейса IpUI, который посылает по указанному адресу заказанный абонентом контент. Данный метод позволяет послать абоненту как SMS-, так и мультимедийное сообщение в зависимости от возможностей его терминала.
 - 24) Сервисный интерфейс вызовом метода sendInfoRes() возвращает результат взаимодействия с абонентом на свой интерфейс обратной связи.
 - 25) Такое сообщение используется прикладным интерфейсом IpAppUI обратной связи для передачи предыдущего сообщения интерфейсу приложения.
 - 26) Приложение вызывает метод release() интерфейса IpUI для освобождения UI объекта.
 - 27) Приложение вызывает метод release() интерфейса IpCall для освобождения Call объекта.
- Корректность и работоспособность данной модели проверены путем запуска ее программной реализации на Java для обслуживания одного абонента в программе моделирования Parlay-шлюза "Ericsson OSA/Parlay simulator". В процессе разработки услуги были выявлены следующие особенности использования открытых интерфейсов Parlay:
- для разработки Parlay-услуг и изучения спецификаций интерфейсов Parlay необходимо помимо выбранного языка программирования владеть технологиями UML, IDL, WSDL и CORBA;

– разработать Parlay–услугу, воспользовавшись исключительно спецификациями интерфейсов Parlay затруднительно из–за недоступности сред разработки, использующих так называемые «чистые» (pure) Parlay–методы, вследствие чего разработчику приходится обращаться к какой–нибудь определенной среде разработки – SDK (Service Development Kit), дополнительно изучать ее спецификации и ограничиваться в разработке возможностями конкретного SDK;

– конфигурировать и тестировать разработанную в SDK услугу можно либо средствами, встроенными в SDK, либо специальными средствами тестирования;

– конфигурировать и тестировать разработанную pure Parlay–услугу необходимо в применении к непосредственно функционирующей операторской платформе, включающей интерфейс Parlay.

В заключение отметим следующее. Использование открытых интерфейсов прикладного программирования обеспечивает стандартизацию доступа VAS услуг внешних разработчиков и сервис–провайдеров к ресурсам базовой сети, необходимым для работы указанных услуг.

Использование открытых интерфейсов прикладного программирования и архитектуры OSA/Parlay формируют единую базу в виде спецификаций интерфейсов Parlay, опубликованных в виде рекомендаций ETSI и 3GPP, для разработчиков VAS услуг и ускоряет появление инструментальных средств для создания новых услуг.

Следует рассмотреть вопрос о внесении спецификаций интерфейсов Parlay в национальные стандарты Украины, что сформирует единую законодательную базу, определяющую технико–экономические условия внедрения новых услуг и тем самым ускорит привлечение украинских разработчиков к созданию и внедрению новых услуг связи.

Разработанная модель услуги наглядно показывает независимость от протоколов базовой сети, так как при создании услуги не учитывались протоколы нижележащих сетей (в данном случае GSM, CDMA или UMTS), а использовался стандартный доступ VAS услуг к ресурсам базовой сети. Именно в этом легкость и удобство разработки VAS услуг для архитектуры OSA/Parlay. В противном случае услугу пришлось бы программировать в отдельности для каждого типа мобильной сети. В свою очередь, построение модели с использованием языка UML значительно ускоряет процесс реализации VAS услуги.

Литература

1. *Закиров З.Г.* Сотовая связь стандарта GSM. Современное состояние, переход к сетям третьего поколения / Факиров З.Г., Надеев А.Ф., Файзуллин Р.Р.; Казанск. гос. тех. унив. – М.: ЭКОТРЕНДЗ, 2004. – 264 с.: ил., табл. – Библиогр.: с. 256-260.
2. *Гибридные сети: как их строить* [Электронный ресурс] / М.А. Шнепс–Шнеппе // Электросвязь.– 2001, № 1.– С. 40. – Режим доступа до журн.: http://www.svetets.ru/paper/01_01.html.
3. *Архитектура Parlay и NGN* / Шнепс–Шнеппе М.А., Килин А., Родина М., Чистов А., Нами от Д. – М.: ЭКОТРЕНДЗ, 2004. – 137 с.: ил., табл. – Библиогр.: с. 134-135.
4. *The Parlay Group / Parlay specifications 6.0.* – Режим доступа: <http://www.parlay.org/specs/index.asp>. – Заголовок з экрана.
5. *Сети нового поколения NGN* [Электронный ресурс] / М.А. Шнепс–Шнеппе // Вестник Связи. – 2002, № 7. – С. 42. – Режим доступа до журн.: http://www.svetets.ru/paper/07_02.html.
6. *Кафедра "Сотовика" / OSA/Parlay: Лекция 4. Подробнее о Parlay-интерфейсах (для программистов).*– Режим доступа: http://sotovik.ru/lib/news_article/news_26620.html. - Заголовок з экрана.
7. *Сети нового поколения и протокол сигнализации SIP* [Электронный ресурс] / М.А. Шнепс–Шнеппе // Электросвязь. – 2002, № 9. – С. 40. – Режим доступа до журн.: http://www.svetets.ru/paper/09_02_3.html.
8. *Барач И.* Окружение для создания услуг. – Режим доступа: http://www.ericsson.com/hr/etk/revija/Br_2_2005_RU/okruzenje.pdf. – Заголовок з экрана.