



МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ КОМПОНЕНТ НА ОСНОВЕ TAPI ТЕХНОЛОГИИ

САЕНКО А.В.

Рассматриваются вопросы формализации процессов взаимодействия компонент программного обеспечения систем передачи данных по модемным каналам при использовании TAPI технологии. Предлагается структура расширенной модели, рассматриваются вопросы ее реализации.

1. Актуальность

В современных телекоммуникационных системах существует проблема создания программных компонент (приложений), обеспечивающих эффективную передачу и прием данных. Понятие эффективности непосредственно связано с необходимостью учета специфики как самого устройства и операционной системы, так и среды передачи управляющей информации [1]. Создание специализированных приложений под конкретные виды устройств превращается в трудоемкий процесс отладки и тестирования.

Выходом из данной ситуации является создание универсальной модели взаимодействия компонентов телекоммуникационной системы, которая должна дополнительно отображать как элементы самой технологии, так и уровни операционной системы.

2. TAPI технология

На сегодняшний день среди большого числа различных технологий можно выделить TAPI (Telephone / Transport Application Programming Interface). Это один из самых важных и наиболее часто используемых наборов функций, созданных фирмой Microsoft. Назначение TAPI – дать возможность программистам создавать приложения, которые работают вне зависимости от физической среды передачи данных и ее физических характеристик, для доступа ко всем свойствам телефонных сервисов в рамках операционных систем Windows. Технология, реализованная данным набором функций, позволяет программистам управлять и манипулировать любым типом коммуникационной связи между компьютером и телефонной линией. Для пользователя же ценность TAPI заключается в том, что он может устанавливать TAPI-совместимое программное обеспечение на системы с различны-

ми физическими типами линий и корректно с ним работать.

Таким образом, TAPI технология идеально подходит для создания моделей взаимодействия компонент телекоммуникационных систем. Для правильного понимания роли TAPI в модели необходимо также рассмотреть ее архитектуру и роль в операционной системе.

3. TAPI архитектура

Для корректного использования реализованных в TAPI возможностей важно понимать ее архитектуру, которая предназначена для предоставления доступа к телефонным сервисам на всех Windows платформах. В ней можно выделить две основные компоненты: устройства и сервисы.

Понятия устройства является фундаментальным. Оно предназначено для абстрагирования от технического обеспечения, которое предоставляет телефонные сервисы, за счет реализации универсального интерфейса доступа к различным типам коммуникационных устройств [2]. Это стало возможным, с одной стороны, благодаря тому, что большинство типов коммуникационных каналов связи оперирует такими понятиями как вызов, ответ на звонок, соединение, разрыв линии и т.д. С другой стороны, поставщики технического обеспечения реализуют TAPI совместимый SPI (Service Provider Interface) интерфейс. В его задачи входит взаимодействие с конечным устройством через его драйвер и трансляция в конкретные вызовы функций общего интерфейса. SPI интерфейс также ответствен за корректное взаимодействие с расширенными возможностями оборудования, не доступные другим типам линий. Он обеспечивает механизм их опроса и активизации.

Сервисы TAPI архитектуры представляют иерархию функций взаимодействия с конечным техническим обеспечением. Под конечным сервисом подразумевается набор определенных функций (вызовов), которые связаны между собой и реализуют ту или иную функциональность коммуникационного приложения. Выбор группы из данной иерархии определяется задачами программирования, а также возможностями и расширениями самого оборудования.

4. Устройства TAPI архитектуры

В архитектуре TAPI определены следующие два класса устройств: устройство-линия и устройство-телефон. Согласно данной классификации, TAPI имеет два независимых набора функций, реализующих поддержку соответствующего класса. Каждый набор сосредоточен на том, что именно подразумевается под устройством.

В TAPI модели устройство-линия не обязательно соответствует физическому устройству. Это просто объект, представляющий физическую линию или, что важнее, ее какое-то отдельное свойство. TAPI

приложение может работать с несколькими устройствами-линиями, каждое из которых соответствует физическому устройству. То же самое приложение может работать с устройствами-линиями, количество которых больше количества физических устройств, доступных системе. Примером этому могут послужить обыкновенные и ISDN модемы. Обычный модем, поскольку он имеет только один канал, представляется в TAPI одним устройством-линией, в то время как один ISDN модем представляется двумя устройствами-линиями.

Устройства-телефоны предназначены для моделирования “виртуальных телефонов” на рабочих станциях. Компьютер со звуковой картой, наушниками и микрофоном может эмулировать все функции стандартного телефона. Виртуальные телефоны, как и их аналоги устройства-линии, необязательно существуют в прямом соответствии с физическими устройствами. Один компьютер может имитировать несколько устройств-телефонов, каждый из которых будет иметь свои собственные характеристики. Следует, однако, заметить, что устройства-телефоны при работе используют устройства-линии, которые, так или иначе, связаны с физическими линиями.

Использование того или иного класса устройств определяется задачами, стоящими перед создателями программного обеспечения. Если реализовываются высокоэффективные коммуникационные приложения, осуществляющие полный контроль состояния линии передачи и потока данных, то необходимо использовать устройства-линии. Если же задача заключается в создании офисных приложений, предоставляющих пользователю удобный интерфейс инициирования звонков и ответа на них, то выбор, как правило, останавливается на устройствах-телефонах. Следует заметить, что процент использования устройств-линий значительно выше использования устройств-телефонов.

Применение в архитектуре TAPI понятия устройства позволяет осуществить гибкий доступ к линиям различного типа. К гибкости доступа относят:

1. Динамическое подключение линий с разным потоком данных, т.е. возможность совместного использования нескольких устройств-линий при наличии одной физической линии. Например, TAPI приложение может предоставлять пользователю передачу данных, факса и голосовых данных. Оно определяет три устройства-линии: для обычных телефонных звонков, для передачи факса, для передачи и приема данных через подключенный модем. Каждый раз, когда TAPI приложение начинает использовать устройство-линию, оно неявно делает запрос первой доступной физической линии на предмет наличия необходимых свойств (передача данных, факса и прочее). Если линия по каким-то причинам недоступна, приложению возвращается соответствующее сообщение. Если доступны

сразу несколько линий, то TAPI приложение использует их по мере необходимости.

2. Динамический выбор разнородных физических линий, т.е. возможность отслеживать типы доступных линий (высоко- и низкоскоростных). Например, одна из подключенных к системе физических линий может быть высокоскоростной линией передачи данных, а другая — обычной линией передачи голосовых данных. Если TAPI приложение сделало запрос на высокоскоростную линию, то SPI вернет ему ее дескриптор. Если приложение сделало запрос на обычную линию, то SPI вернет приложению дескриптор обычной линии. Если же приложение сделает запрос на высокоскоростную линию, в то время как она уже используется, то SPI, зная, что обычная линия не обладает требуемыми свойствами, вернет сообщение о том, что все линии заняты.

5. Сервисы TAPI архитектуры

Для TAPI технологии, имеющей широкий набор функций, существуют взаимовязанные четыре логические группы (рис. 1):

- сервисы вспомогательной телефонии;
- сервисы основной телефонии;
- сервисы дополнительной телефонии;
- сервисы расширенной телефонии.

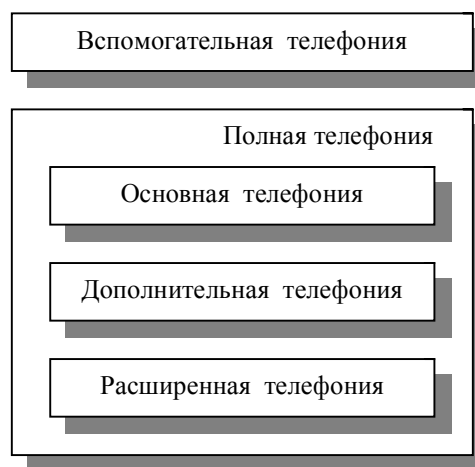


Рис.1. Структура сервисов TAPI

Разбиение на группы осуществляется на основе решаемого класса задач. Логическое деление функций на уровни и их содержание одинаково для устройств-линий и устройств-телефонов в терминах TAPI. Сервисы основной, дополнительной и расширенной телефонии нередко называют полной телефонией.

Сервисы вспомогательной телефонии. Функции данной группы предоставляют приложению возможность реализовывать исходящие звонки и определять настройки локализации, настраиваемые, как правило, пользователем при установке операционной системы (код страны, код города и пр.). Этот набор функций используется для класса задач,

требующих инициировать исходящие звонки. В качестве примера можно привести поставляемую с операционной системой Windows программу dialer (номеронабиратель). Следует заметить, что функции данного сервиса не дают возможности осуществлять мониторинг изменения состояния линии, контролировать процесс приема/передачи данных и пр.

Сервисы основной телефонии. Функции данной группы являются минимальным набором спецификации TAPI. Эти функции обеспечивают совместимость приложения с любым провайдером сервисов. Реализованная данной группой функциональность соответствует классам задач, поддерживающих работу с обычными телефонными линиями. Хотя API основной телефонии может быть использовано и для T1 или ISDN цифровых линий, расширенные сервисы этих линий, такие как организация конференций, недоступны при использовании функций основной телефонии.

Сервисы дополнительной телефонии. Функции данной группы соответствуют классу задач, поддерживающих дополнительные функции телефонии, а именно: организация конференций, переадресация звонка, помещение его в очередь и пр. Сервисы дополнительной телефонии являются необязательными для провайдера сервисов, т.е. он сам решает, какие ему сервисы использовать, а какие нет. В связи с этим, приложение, перед тем как применить то или иное устройство, должно опросить его на предмет поддерживаемых функций.

Особенностью сервисов дополнительной телефонии является то, что конкретное поведение сервиса определяет TAPI, а не провайдер сервиса. Это значит, что провайдер сервиса должен реализовывать поддержку какого-либо сервиса лишь в том случае, если сервис по логике работы соответствует представлениям о нем TAPI. Если это не так, то сервис должен быть представлен в сервисах расширенной телефонии.

Сервисы расширенной телефонии. Это набор функций, предназначенных для класса задач, поддерживающих нестандартные функции телефонии. Такие функции производители технического обеспечения могут добавлять сами и при этом работать в рамках модели TAPI сервисов. Поскольку API предоставляет лишь механизм получения сведений о расширенных сервисах, то их интерпретация целиком зависит от провайдера сервиса и самого приложения.

6. Обеспечение соответствия стандартам архитектуры операционной системы

Технология TAPI разработана в рамках архитектуры операционной системы Windows, называемой WOSA. Концепция WOSA (Windows Open Service Architecture) модели заключается в предоставлении доступа к расширенным сервисам операционной системы Windows, требуя при этом лишь со стороны

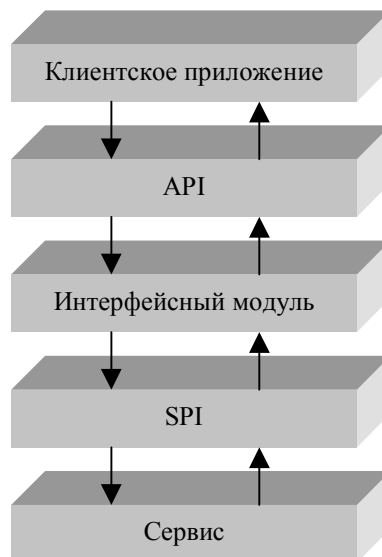


Рис. 2. Взаимодействие клиентов с системными сервисами в WOSA модели

пользовательского приложения минимальный набор информации о самом сервисе. Для обеспечения гибкости WOSA модель определяет два интерфейса – API клиентской стороны и SPI сервиса. Они соединяются единым интерфейсным модулем, который взаимодействует с API и SPI приложениями. Клиентское приложение должно соответствовать требованиям API и

интерфейсного модуля, а все сервисные приложения должны соответствовать требованиям SPI и интерфейсного модуля (рис. 2).

Каждое клиентское API обеспечивает единый набор вызовов для доступа к сервисам. Основным моментом является то, что клиентское приложение имеет возможность лишь запрашивать определенные сервисы, но никак не может напрямую взаимодействовать с ними. Даже запрос посылается не прямо сервисной части, а интерфейсному модулю. Сервис через SPI принимает запросы и функционирует в соответствии с ними. Большинство SPI приложений располагаются на серверах в сети или реализуются на рабочих станциях как независимо выполняемые приложения. SPI, как правило, не предназначено для взаимодействия напрямую с клиентским приложением. Запросы на сервисы поступают от интерфейсного модуля, и SPI возвращает информацию интерфейсному модулю, который впоследствии через API передает ее клиентскому приложению. Сервисная часть разрабатывается так, чтобы обеспечивать одновременное обслуживание нескольких клиентов. При этом за корректное взаимодействие с каждым из клиентов ответственен сам SPI.

Поскольку ключевым аспектом в создании WOSA модели является изоляция API клиента и SPI сервера, то требуется интерфейс взаимодействия между этими двумя компонентами. Он реализуется в виде динамически загружаемой библиотеки (DLL), которая позволяет подключаться к существующим сервисам на этапе выполнения, а не на этапе компиляции. Таким образом, имеется возможность изменять интерфейсный модуль, не внося изменений в сами клиентские приложения. До внедрения WOSA модели интерфейсные модули создавались для обеспечения взаимодействия между приложением и конечным сервисом. Другими словами, интерфейсный модуль “знал”, как обращаться с

конкретной версией сервиса. В настоящее время интерфейсные модули могут взаимодействовать с любым провайдером сервисов, который поддерживает набор SPI вызовов. Один такой интерфейсный модуль может связать приложения с любой версией сервиса или сразу с несколькими сервисами.

Являясь частью WOSA модели, TAPI обеспечивает полную реализацию телефонных сервисов для операционных систем, изолируя приложение от специфических вызовов поставщика технического обеспечения. Разработчики приложения могут сосредотачиваться на разработке необходимых свойств приложения и оставить специфичную для поставщика реализацию программистам SPI интерфейса. В то же время, поставщики технического обеспечения реализуют набор SPI вызовов, который гарантированно будет работать на всех Windows платформах.

7. Модель взаимодействия компонент на основе TAPI технологии

В предлагаемой модели взаимодействия (рис.3) выделим три типа обрабатываемых данных: поток данных, представляющий любую информацию, данные от факса и голосовые данные. Этими тремя типами можно ограничиться как вполне исчерпывающими. Модель основана на клиент/серверной технологии. При этом клиент выступает в роли инициатора звонков и передает, для простоты, один из перечисленных выше типов данных — поток информации от модема. Сервер принимает звонки от трех типов устройств: факса, модема и телефона, которые он обрабатывает по требуемой от программного обеспечения логике. Например, данные от модема сохранить, данные от факса передать соответствующей коммуникационной программе, а звонок от телефона занести в журнал событий.

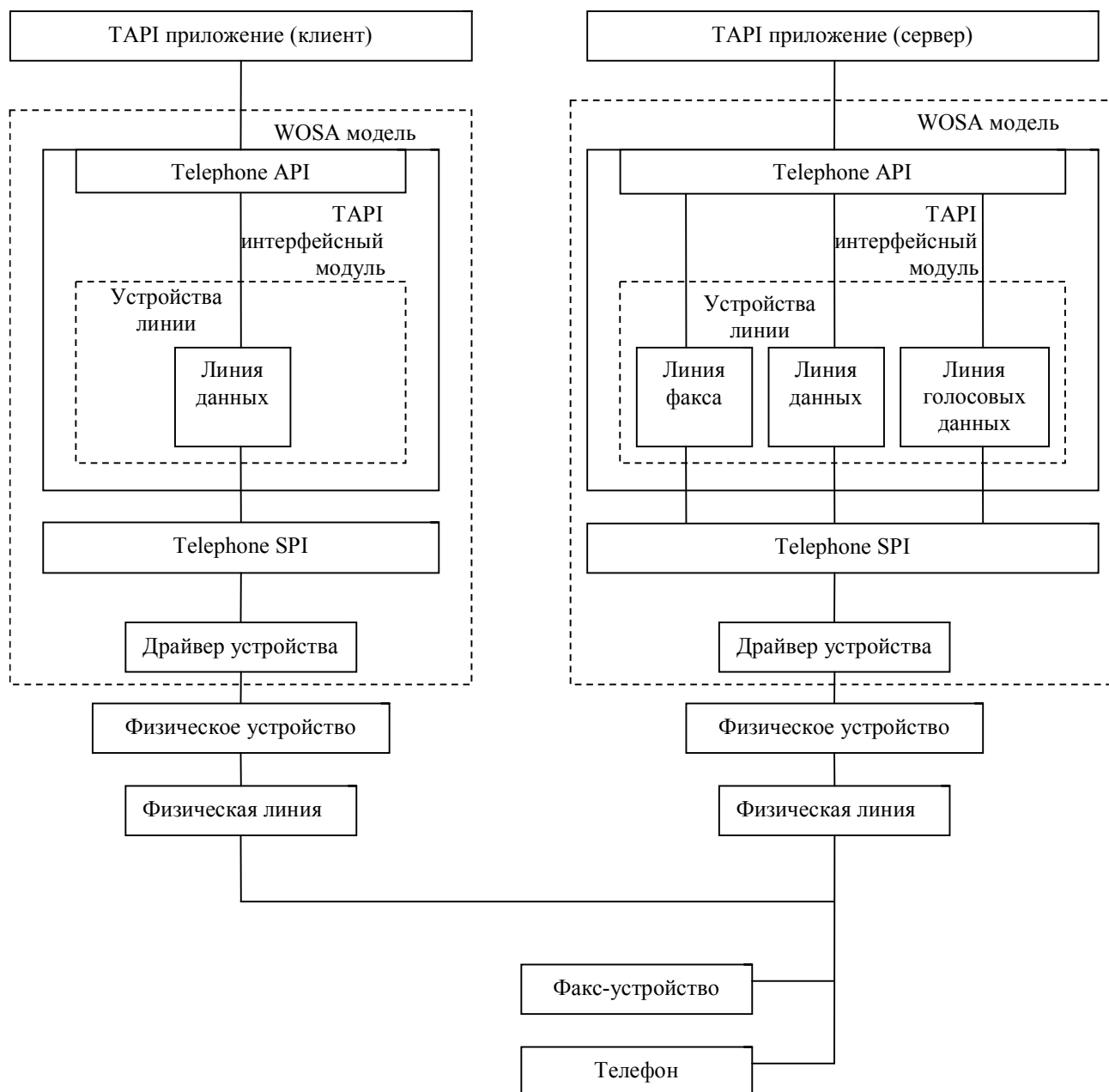


Рис. 3. Предлагаемая модель взаимодействия компонент телекоммуникационной системы

Звонок от клиента серверу осуществляется функционирующим на нем клиентским приложением. Для этого клиент использует интерфейс прикладного программирования (API), предоставленный технологией TAPI. Так как подразумевается, что клиент будет передавать данные по модему, то он сначала делает запрос на соответствующий тип линии. Если такая линия есть и она свободна, то клиентское приложение получает ее дескриптор. Процесс инициирования звонка заключается в вызове соответствующей функции библиотеки TAPI. Данный запрос через интерфейсный модуль передается провайдеру соответствующего телефонного сервиса (TSPI), который, в свою очередь, передает его драйверу установленного в системе модема. При установлении соединения драйвер сообщит об этом провайдеру интерфейса и передаст дополнительную информацию – скорость, протокол. Провайдер интерфейса через соответствующий вызов функции прикладного программирования передаст информацию клиентскому приложению. После этого клиент начинает отправлять данные. По окончании сеанса клиент закрывает линию посредством вызова соответствующей функции, которая интерпретируется в конце драйвером модема как команда к разрыву соединения.

На серверной части выполняются подобные операции. За прием и обработку звонков ответственно функционирующее серверное программное обеспечение. Прежде, через вызовы функций прикладного программирования оно получает дескрипторы соответствующих типов линий и дает команду перейти в режим ожидания появления звонка. При поступлении входного звонка драйвер устройства сообщает об этом провайдеру интерфейса (TSPI), который через механизм сообщений передает информацию серверному приложению.

Структура сервера и клиента представлена в соответствии со структурой, которую образуют совместно уровни TAPI технологии и операционной системы.

8. Заключение

В работе показана необходимость создания модели взаимодействия компонентов для телекоммуникационной системы и выбора технологии ее реализации. На основании проведенного анализа предъявляемых требований была выбрана технология TAPI. Рассмотрены ее архитектурные особенности, делающие данную технологию универсальной по отношению к разным типам коммуникационного оборудования. Также дано представление о структуре ее сервисов. Значительное внимание было уделено месту технологии TAPI в семействе операционных систем Windows с подробным описанием архитектуры открытых сервисов (WOSA). На основании изложенных материалов была предложена модель взаимодействия компонентов с подробным описанием ее составляющих и принципов функционирования.

Литература: 1. *Саенко В.И.* Администрирование, управление и мониторинг в компьютерных сетях // АСУ и приборы автоматики. 1998. № 108. С.251-258. 2. *Michael Amundsen.* MAPI, TAPI, SAPI Developer's Guide. SAMS, 1996. 600 p.

Поступила в редколлегию 23.10.2000

Рецензент: д-р техн. наук, проф. Левыкин В.М.

Саенко Александр Владимирович, студент факультета КН ХТУРЭ. Научные интересы: сетевые технологии, администрирование и удаленный доступ к ресурсам. Хобби и увлечение: моделирование, авиация. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.