

АННА БИТЕЛЕВА

Еще раз о middleware

Middleware (MW) в системах IPTV — понятие столь же важное, сколь и расплывчатое. Поэтому материалы о плюсах и минусах разных подходов к реализации MW мы решили предварить вступительной статьей. Легко обнаружить, что этот термин используется для обозначения совершенно разных программных компонентов системы.

Следует выделить две категории ПО, обозначаемых этим термином. К первой относится ПО, ответственное за координацию работы всех программно-аппаратных компонентов системы. Функции такого MW обычно иллюстрируются картинкой, аналогичной представленной на рис 1. Часть модулей, координируемых таким MW, входит в состав решения, а часть предоставляется сторонними производителями. Такие комплексные решения предлагают Thomson, SeaChange, Espial, Microsoft, Ericsson и многие другие компании.

Но в этих материалах пойдет речь о MW в более узком смысле этого слова. А именно о ПО, реализующем интерактивные приложения и пользовательский интерфейс в абонентских приставках. Для решения

этой задачи также существуют два принципиально разных подхода. В первом случае сами приложения реализуются на головных серверах, а приставка берет на себя только задачу вывода результата на экран с помощью встроенного в нее браузера. Условно назовем эти системы «браузерными». В приставках могут использоваться усеченные версии стандартных браузеров, таких как Opera или Mozilla, или же браузеры, специально разработанные для IPTV-приставок, например, компания ANT. Они штатно интегрируются в приставки их производителями. В этом случае термином MW обозначают модули головного ПО, ответственные за реализацию этих приложений и являющиеся частью MW в широком смысле этого слова.

Во втором случае обработка приложений выполняется внутри самой приставки. Этот подход требует серьезного вмешательства в ее ПО. Из приставки удаляется все до уровня операционной системы и пишется новое ПО. Структура ПО такой приставки (вернее, слои ПО, независимые от ее аппаратной платформы) изображена на рис. 2.

В таких решениях под MW понимается оболочка, позволяющая интегрировать в нее приложения. Она размещается между приложениями и операционной системой приставки. Это, собственно, то, что понимается под MW в компьютерном мире, откуда пришел сам термин. Поэтому многими специалистами в области IT такое использование термина воспринимается как единственно правильное.

Мы не ставим перед собой задачи бороться за чистоту терминологии, а просто хотим обрисовать особенности каждого подхода.

В «браузерной» системе головное приложение должно получить запрос от приставки, собрать на сервере нужные данные и оформить их в виде HTML-страницы, которая будет воспроизведена браузером, или отправить видео, которое будет воспроизведено плеером. Создание таких приложений не представляет особой сложности, так как вся обработка осуществляется на серверах, а представление результата через браузер является достаточно стандартной задачей. Надо просто сформировать HTML-страницу и реализуемые с этой страницы скрипты¹, написанные, как правило, на JavaScript. Интеграция приложения с приставкой заключается в организации его взаимодействия с API (Application Programming Interface) приставки. API обеспечивает программное взаимодействие ПО приставки с внешними устройствами. В частности, оно определяет команды запроса, приема и воспроизведения приложения браузером или плеером. Кроме того, для «браузерных» приложений должна быть проверена корректность воспроизведения страницы браузером.

Иногда возникают сложности с исполнением скриптов. Они обусловлены некоторыми различиями в представлении браузерами объектной структуры документа (DOM)². Но в целом адаптация к конкретному браузеру выполняется достаточно легко.

¹ Скрипт — это интерпретируемая программа, реализующая какой-то сценарий (последовательность действий), в рассматриваемых случаях инициируемая через элемент пользовательского интерфейса.

² Это формальное описание элементов документа, в данном случае страницы браузера, а также описание взаимоотношений этих элементов, реализованное в виде дерева. В разных браузерах это может быть сделано немного по-разному.

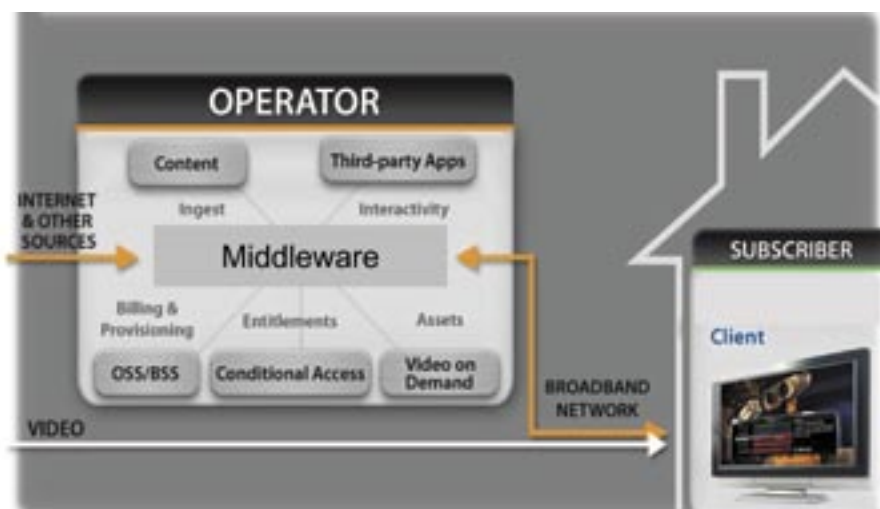


Рис. 1. MW как ядро комплексной системы IPTV

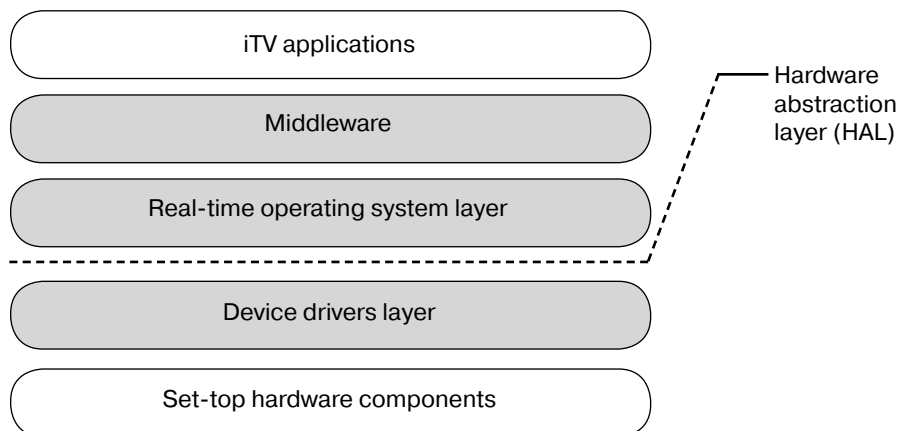


Рис. 1. Многослойная структура ПО абонентской приставки

Кроме того, большинство современных браузеров поддерживает разные форматы векторной графики, например SVG, при достаточных аппаратных ресурсах приставки в них можно использовать такие браузеры. Они добавляют дополнительные графические возможности и обеспечивают плавность перемещения элементов внутри картинки. В любом случае, приложения пишутся на широко распространенных открытых языках.

Второй подход, то есть реализация обработки приложения в самой приставке, требует серьезного вмешательства в ее ПО со стороны программистов, которые разрабатывают для приставки MW в компьютерном смысле этого слова, а также конкретные приложения, работающих поверх этой MW.

Перепрограммирование приставки — достаточно сложная работа, требующая взаимодействия с ее операционной системой (а иногда изменяется и сама ОС), для проведения которой необходимо получить SDK³ приставки у ее производителя. Соответственно, стоимость самой приставки повышается на стоимость SDK и работы по перепрограммированию. Очевидно, что добавление новых приложений при такой схеме также выливается в дополнительное программирование, которое может быть выполнено только разработчиком конкретного MW.

Что касается исходных требований к аппаратной начинке приставки, то установка MW и приложений в большинстве случаев не требует дополнительных ресурсов по сравнению с браузерным отображением.

Существующие стандарты на MW для приставок, в первую очередь MHP и появившийся в качестве ее развития GEM (Globally Executable MHP), оптимизированы для применения в сетях DVB. Оптимизация заключается в том, что они работают с механизмами передачи данных для интерактивных приложений, заложенными в DVB.

В то же время, в IP-среде можно передавать эти данные более простыми способами. Кроме того, эти MW достаточно сильно загружают ресурсы приставок. Поэтому в сетях IPTV применяются корпоративные, нестандартизированные решения со всеми вытекающими минусами.

Однако у этого подхода есть и существенные преимущества перед «браузерным» — более эффективное использование ресурсов приставки, минимизация трафика между приставкой и серверами и, самое главное, — быстродействие. Например, вызов EPG в «браузерной» системе потребует обработки запроса соответствующим скриптом и отправки его удаленному серверу EPG, который после идентификации приставки сформирует нужную страницу и вернет ее приставке. Причем если в буфере сервера не окажется нужных данных, то он должен еще получить их из базы, размещенной на диске другого сервера. Весь процесс может затянуться на несколько секунд.

Кроме того, очевидно, что применение браузеров сопровождается значительным трафиком, особенно от сервера к приставке. Ведь каждое изменение вида пользовательского интерфейса требует получения новой картинки с сервера. Во всяком случае, так было в ранних браузерных системах.

В современных решениях используются механизмы, снижающие объемы передачи. В первую очередь это технология AJAX (Asynchronous JavaScript and XML), созданная для интерактивных web-приложений. Используясь совместно с HTML и Script, она позволяет не запрашивать у сервера изображение всего интерфейса, а получать информацию только об изменениях, причем делать это в режиме, не привязанном к выводу картинки на экран. AJAX в некоторых случаях позволяет снизить трафик обмена между сервером и приставкой. Но так как его использова-

ние сопровождается увеличением числа передаваемых с сервера скриптов, то применение этой технологии не всегда приводит к желаемому результату.

Еще один способ снижения объемов передачи от сервера к клиенту заключается в кэшировании страниц или элементов изображения, повторяющихся от страницы к странице.

Тем не менее, полностью проблемы браузерного отображения они не снимают. Поэтому многие компании предлагают комбинированные решения, в которых часть приложений вынесена на сервер, а часть реализуется в приставках. В частности, комбинированные решения имеют уже упоминавшаяся ANT и компания «Нетрис», материал которой опубликован в этом номере.

Ниже вниманию читателей предлагаются три материала, рассматривающие плюсы и минусы различных подходов к реализации MW. Первой в редакцию была прислана статья из компании «НетАП». Достоинства и недостатки обоих подходов рассмотрены в ней с точки зрения разработчика клиентской MW, который хорошо знаком с достоинствами своего подхода и хуже — с возможностями браузерных систем. Поэтому для баланса мы попросили прокомментировать эту статью разработчиков систем, работающих с web-технологиями. Эта статья была написана с целью объяснить оба принципа немного подробнее и привести более полную классификацию решений для организации интерактивных приложений. Надеемся, что коллективный портрет MW, отображенный в четырех представленных материалах, получился достаточно полным и объективным.

³ SDK (Service Development Kit) — информация о программных интерфейсах внутреннего ПО приставки, необходимая для его модификации.



WINERSAT®
The name you can count on

WXM-860SL

- 2 QPSK/мультиплексор/QAM модулятор/RF up converter
- Фильтрация PID, удаление/добавление программ в выходной поток, редактирование PID и т.д.
- QAM IF вход/выход.

А также весь спектр профессионального цифрового оборудования:

- DVB-S/S2 приемники
- MPEG-2 кодеры
- Мультиплексоры
- QAM-модуляторы
- Трансмодуляторы и др.

www.ditel-telecom.ru

Официальный представитель компании Winersat в России - **DiTel Ltd.**
Россия, 198095, С.-Петербург, Митрофаньевское ш. д. 5Е, тел. +7 (812) 655-5052. E-mail: post@ditel-telecom.ru

Экономичное и качественное решение для начала и развития цифрового вещания в сети кабельного телевидения

СУПЕРЦЕНА на WXM-860SL

Звоните: +7 (812) 655-50-52



Сертификат
Министерства Связи
на все оборудование

NEW!



WXM-860SL

4 устройства в 1 корпусе:
2×DVB-S/S2 + 2×ASI ⇔ QAM ⇔ RF (47-862МГц)



Digital Telecommunications

Подходы к реализации IPTV middleware

АНТОН ГАВРИКОВ

компания «НетАП»

Существуют два основных и совершенно разных подхода реализации системы middleware. Первый основан на web-технологиях. В этом случае на приставке работает браузер, предлагаемый (за отдельную плату) заводом-изготовителем. Он отображает генерируемую сервером web-страницу пользовательского интерфейса. Средства взаимодействия с плеером — java script. Второй подход заключается в том, что с приставки удаляется заводской софт, устанавливается операционная система, под которой графический интерфейс управления интерактивным ТВ — это уже отдельное, самостоятельное нативное¹ приложение.

Middleware на web-технологиях. Первое поколение

Графический интерфейс абонента в такой реализации представляет собой набор web-шаблонов. Они хранятся на сервере и передаются на клиентскую приставку при каждом запросе каждого абонента. На клиентской приставке запущен браузер, с помощью которого и отображается необходимая web-страница. Будь то список ТВ-каналов, страница услуги VoD или информация о балансе — для каждой страницы необходим свой web-шаблон.

Считается, что разработать и модифицировать такой интерфейс сравнительно просто. Но на практике возникает множество проблем, сводящих к минимуму достоинства. Ниже перечислены основные недостатки.

1. Несовместимость разных приставок внутри одной сети. Это происходит из-за того, что в разных приставках установлены разные браузеры. А каждый браузер по-своему отображает одну и ту же страницу. У каждого свой API для Java Script. Таким образом, web-интерфейс путем сложных манипуляций «подгоняется» под требования какого-либо конкретного браузера.

2. Невысокая скорость отображения интерфейса. Возникает из-за излишне длинной цепочки получения конечного результата. Запрос от абонента поступает на сервер. Сервер генерирует необходимую web-страницу, после чего она скачивается и обрабатывается браузером. При нагруженной сети эти задержки могут увеличиваться. Стоит учесть, что и сам браузер является неэффективным потребителем системных ресурсов приставки, что затормаживает работу интерфейса в целом.

3. Излишняя загруженность серверной части. Необходимость предоставлять для каждого пользователя свою страничку значительно нагружает сервер системы middleware.

При таком подходе в итоге оператор получает систему, которую будет очень сложно модифицировать. Технологии не стоят на месте, на рынке постоянно появляются новые и качественные ТВ-приставки по заманчивым ценам. Но использовать эти новинки в системе middleware первого поколения, скорее всего, не получится. У абонента же — сомнительной скорости и функциональности интерфейс, работа с которым может вызвать вопросы о целесообразности всего интерактивного телевидения в принципе.

Middleware с низкоуровневой интеграцией приставок. Второе поколение

Одной из первых в России перешла к использованию нативных приложений (толстый клиент) для построения системы middleware компания «НетАП». Преимуществом нативного приложения является возможность работы практически непосредственно с аппаратными ресурсами. Для низкоуровневой интеграции приставок с IPTV-комплексом требуются партнерские отношения с производителями телевизионных приставок, так как именно SDK от производителя позволяет использовать ресурсы STB максимально корректно и наиболее полно.

В middleware второго поколения задача формирования графического интерфейса полностью переложена на клиентскую часть системы. На STB устанавливается операционная система

Linux, под управлением которой работает специальное приложение, являющееся самостоятельным интерфейсом пользователя. То есть вся графика, все подпрограммы, модули и плагины находятся и исполняются на самой приставке, не обращаясь, без необходимости, к серверу. Это позволяет не только разгрузить сервер, но и предоставить еще ряд других значительных преимуществ:

1. В одной сети можно использовать разные приставки. Это стало возможно благодаря тому, что одно и то же приложение интерфейса собирается для каждой конкретной приставки, используя SDK от производителя. А протокол взаимодействия между сервером и клиентом един для всех типов приставок. Любая новая приставка может быть подключена к сети сразу же, как только для нее будет собрана своя прошивка.

2. Быстрота работы. Если нет необходимости подгружать недостающую графику и обрабатывать разнообразные скрипты, то нет и задержек при «перелистывании» страниц доступа к разнообразным услугам.

3. Автономность. В случае сбоя серверного оборудования системы middleware абонент сможет продолжать получать медиаконтент в течение всего оплаченного срока подписки на услуги.

4. Отсутствуют ограничения визуализации и функциональности. Любая задумка может быть воплощена без особых проблем. Если нет браузера и шаблонов web-страниц, значит нет и никаких ограничений в реализации графики или функционала.

5. Автоматическое обновление прошивок. Если возникает желание внедрить в IPTV-комплекс новый сервис, то следует только запустить автоматическое обновление софта на клиентской приставке. Новые ресурсы станут доступны всем.

Таким образом, при использовании системы middleware второго поколения оператор связи получает решение с неограниченными возможностями. Новые приставки и новые сервисы могут быть введены в эксплуатацию без остановки всего комплекса IPTV. Абонент же оценит быстроту работы графического интерфейса и простоту добавления новых сервисов.

¹ Компилированная программа, написанная под аппаратно-программные ресурсы конкретной приставки (прим. ред.)

Комментарии к статье «Подходы к реализации IPTV middleware»

1

К функциям middleware принято относить:

- Интеграцию основных компонентов решения.
- Реализацию пользовательского интерфейса.
- Предоставление инструментов для управления сервисами в системе IPTV.
- Авторизацию пользователей и предварительный биллинг.

Задачи интеграции приставки (STB) с middleware и реализации пользовательского интерфейса в настоящее время находятся в прямой зависимости друг от друга. Исторически выделилось два основных направления развития подобного взаимодействия в IPTV: первое, заимствованное из спутникового телевидения, — через встроенное программное обеспечение приставки, и второе, пришедшее из интернет-технологий, — через web-браузер приставки и сервер на стороне оператора. Данные направления развивались параллельно, и за несколько лет было принято немало попыток нивелировать недостатки и преумножить достоинства каждого из них.

IPTV активно начало развиваться в 2004-2005 годах, когда в освоении широкополосных технологий делались только первые шаги. Производители приставок также находились в поиске подходящих решений. Встроенный web-браузер был логичным ответом на запрос оператора на максимально открытое решение с использованием уже хорошо зарекомендовавших себя web-технологий¹ и возможностью сократить затраты на интеграцию с другими компонентами IPTV. Безусловно, первые браузеры во многом уступали своим компьютерным собратьям, но за несколько лет разработчикам удалось значительно сократить этот разрыв. В настоящий момент при создании интерфейсов возможно применение Flash, Flex, SVG и других форматов, использующих векторную графику, что позволяет разнообразить и оптимизировать дизайн портала. Тем не менее, в погоне за красивой картинкой нужно помнить, что основной задачей пользовательского интерфейса является продажа медиасервисов, и это должно быть краеугольным камнем при разработке портала и оценке его удобства.

Сравнивать два подхода к реализации middleware лучше всего в разрезе нескольких основных аспектов реализации проекта IPTV: интеграция приставки и middleware, сервисы и добавление новых сервисов, интеграция с другими компонентами IPTV и приложениями, стоимость, надежность.

Интеграция приставки и middleware

Интеграция приставки и middleware, использующего web-технологии, происходит с использованием API (Application Programming Interface). Главная задача — реализовать взаимодействие с основными десятью командами, понимаемыми приставкой. Некоторые производители STB используют одинаковые команды; таким образом, интеграция такого middleware с разными приставками занимает относительно короткое время, а при использовании стандартного web-браузера в сети оператора могут работать приставки нескольких производителей. В настоящий момент компания «Нетрис» имеет опыт разработки и внедрения таких решений, где в рамках одной сети одновременно используются приставки нескольких вендоров.

Низкоуровневая интеграция middleware с приставками занимает гораздо больше времени, так как здесь идет работа напрямую с операционной системой приставки. Именно поэтому такой вид интеграции зависит от используемой в STB ОС. Также разработчик middleware должен в этом случае приобрести у производителя приставок SDK (и это обязательно скажется на финальной стоимости), который предоставляет информацию об опыте конструкторов STB для более эффективного использования функций абонентского устройства. Отсутствие стандартов в данной сфере, применение вендорами различных протоколов позволяет говорить о том, что для интеграции с новой приставкой нужно будет повторить весь процесс снова. Тем не менее, игра стоит свеч, так как основное преимущество данного подхода — быстрое действие — может быть получено только при тщательной проработке этого этапа, что сопряжено с серьезными временными затратами.

МУРАД ЭФЕНДИЕВ
технический директор компании «Нетрис»
НАТАЛЬЯ ЯШЕНКОВА
менеджер по маркетингу компании «Нетрис»

Интеграция с другими компонентами IPTV и приложениями

Web-технологии базируются на открытых стандартах, поэтому интеграция с другими компонентами проходит легко и занимает короткое время. Оператор независим от поставщика решения: он может разрабатывать дополнительные приложения собственными силами или заказать их у стороннего производителя и практически без участия разработчика middleware интегрировать их в общее решение.

Для встроенного middleware интеграция и расширение функциональности системы может быть реализовано только силами разработчика middleware, так как программирование осуществляется на системном уровне. Оператор зависим от поставщика решения.

Сервисы и введение новых сервисов

Web-браузер предоставляет несколько преимуществ: абоненты получают встроенный сервис web-серфинга, доступ к популярным интернет-ресурсам, которые имеют адаптированные интерфейсы. Браузер, отвечающий требованиям основных стандартов сети Интернет, является необходимым условием качественного получения таких услуг.

Добавление новых сервисов для встроенного MW происходит на базе механизмов, описанных в предыдущем пункте. Чтобы каждый абонент оператора смог получить обновление для встроенного middleware, приставку нужно перепрошить, что занимает определенное время. При использовании web-браузера — только обновить страницу портала, так как все изменения вносятся только на сервер middleware.

¹ Принцип работы web-браузера в статье описан верно: запрос поступает на сервер, где происходит генерация необходимой web-страницы, после чего она скачивается и обрабатывается браузером. Каждый интернет-пользователь становится свидетелем такой трансформации многие тысячи раз, и при этом данная технология считается успешной.

Комментарии к статье «Подходы к реализации IPTV middleware»

Стоимость и надежность

Для выяснения возможности сократить затраты на приставки при отказе от браузера мы провели опрос ряда ведущих производителей приставок и выяснили: в настоящий момент браузер является неотъемлемой частью приставок и они поставляются только в такой комплектации. Тем не менее, производители приставок могут брать агентскую плату за изготовление встроенных прошивок с использованием SDK (случай встроенного MW).

Преимущество автономности встроенного middleware вызывает сомнение, так как оператор связи должен гарантировать непрерывность и высокое качество предоставляемых услуг, что подразумевает обязательные меры по обеспечению отказоустойчивости отдельных узлов и систем.

Middleware нового поколения

Как было сказано ранее, быстродействие пользовательского интерфейса —

основное преимущество встроенного middleware. Web-браузер предоставляет оператору гибкость, быстроту развертывания системы IPTV, независимость от производителя middleware, простоту добавления новых сервисов. Каждый раз оператор вынужден выбирать между двумя разными подходами, понимая, что остановив выбор на одном из них, он лишается преимуществ другого.

Для того чтобы облегчить выбор оператора, особенно сейчас, в период финансовой нестабильности и оптимизации расходов, компанией «Нетрис» было разработано middleware нового поколения, которое позволяет сочетать быстродействие пользовательского интерфейса и открытость web-технологий. При комбинированном подходе браузер используется для отображения страниц, но «тяжелые» данные (EPG, описание фильмов видео по запросу)

хранятся во внешнем приложении. В сочетании со сбалансированным использованием кэширования страниц такой подход позволяет увеличить скорость обработки страницы в три раза. Полученный нами результат позволяет говорить о быстродействии, соизмеримым с тем, что демонстрирует встроенное middleware.

Реализация пользовательского интерфейса далеко не единственная функция middleware в системах IPTV. Наряду с интеграцией всех компонентов решения, middleware выполняет авторизацию абонентов и осуществляет предварительный биллинг медиауслуг оператора, является основой для внедрения системы медиаметрии. Наш опыт показывает огромную важность качественной реализации всех функций данного компонента решения IPTV для успешного внедрения и развития услуг современного оператора Triple Play.

2

Действительно, можно выделить два подхода к реализации пользовательского интерфейса middleware. Однако разделять их на «первое» и «второе» поколения не совсем корректно. Это два разных подхода, у каждого из которых есть свои плюсы и минусы.

Реализация middleware по принципу «толстого клиента» позволяет сделать пользовательский интерфейс более быстрым по сравнению с HTML-версиями. По этому принципу построены такие зарубежные middleware, как Seachange, Orca. Но стоимость таких решений гораздо выше, нежели решений, основанных на HTML-страницах. Это объясняется высокой стоимостью комплекта средств разработки (SDK) для абонентских приемников, необходимого для создания такого решения (от 10 тысяч долларов США и выше). Как правило, такие решения создаются для одного типа STB (либо для линейки STB одного производителя). И в дальнейшем оператор будет вынужден приобретать приемники только этого производителя.

Middleware, основанные на HTML-страницах, прежде всего рассчитываются на возможность адаптации к новым моделям STB различных производителей без необходимости покупки SDK и

изучения внутренней архитектуры приемника. Интеграция таких middleware к новым приемникам сводится к адаптации HTML-страниц и написанию управляющего кода к браузеру, установленному в абонентский приемник. Большинство современных приемников содержат в себе один из браузеров Mozilla, ANT Galio либо Opera. Поэтому адаптация к новым моделям не занимает много времени.

Действительно, в случае использования HTML-ориентированного middleware нагрузка на серверную часть будет выше, так как с сервера на приемник помимо наборов данных приходится передавать и графические элементы. Однако эта проблема решается грамотно построенной архитектурой middleware, а также возможностями операционной системы, под управлением которой функционирует middleware. Так, например, в middleware eScentra часть, отвечающая за подготовку и передачу страниц на абонентские приемники (портал), может быть вынесена на любое количество спутниковых серверов в зависимости от структуры сети передачи данных оператора, механизм web-фермы (web farm) обеспечит динамическую балансировку нагрузки на узлы, а также отказоустойчивость. К тому же механизмы кэширо-

вания в самом абонентском приемнике позволяют повторно не передавать элементы, которые не были изменены на сервере.

Кроме того, в HTML-ориентированных middleware оператор может самостоятельно изменять некоторые графические элементы интерфейса пользователя (например, логотип, цветовую гамму). Либо даже самостоятельно создавать порталы для новых приемников. Все изменения в этом случае будут автоматически отображены на телевизорах абонентов. В случае же с интеграцией интерфейса пользователя в прошивку приемников оператор будет вынужден обращаться к системному интегратору или к поставщику приставок для получения новой версии прошивки, а затем производить перепрограммирование (обновление) всех приемников в сети. Поэтому и тот, и другой вид middleware имеют право на существование. Выбор между этими типами будет зависеть от условий поставки системы IPTV и ее компонентов, перспектив развития системы и, естественно, бюджета.

АНАТОЛИЙ ФОМИЧЕВ

директор департамента цифровых технологий компании «ТелКо Групп»