



Преимущество сохранения информации на промежуточных носителях для последующей более надежной, достоверной и целостной доставки человечество ощутило еще до изобретения письменности. Какие только носители не использовались! Практически любая подручная материя в твердой фазе может нести информацию. Вот с транспортом сложнее, — как правило, сообщение нес человек, и это не страховало от ошибок и искажений (вспомним хотя бы «Сказку о царе Салтане»).

Международный московский банк (ММБ), основанный в 1989 году, начал осуществлять обмен сообщениями в конце 1990-х — начале 2000-х годов с помощью системы Data Queue (DQ), реализованной на уровне операционной системы IBM OS/400. На сервере AS/400 функционирует основная учетная система банка MIDAS DBA. Поначалу хватало модулей, работающих в среде AS/400, но через некоторое время встала неизбежная проблема интеграции с внешними приложениями, построенными на иных платформах. Однако транзакционное взаимодействие приложений на других платформах с помощью DQ связано с серьезными проблемами, поскольку интерфейс DQ API не имеет средств, управляющих включением блока вызовов функций API в транзакцию.

Поначалу в ММБ система обмена сообщениями IBM Websphere MQ применялась только приложениями на ПК-серверах. По-настоящему крупным приложением, использующим очереди сообщений на базе Websphere MQ, стала система SMSbanking (SMSB), которая состоит из консольных приложений-обработчиков, связанных очередями сообщений. SMSB позволила связать между собой систему приема/отправки SMS-сообщений, систему карточного процессинга, БД телефонов клиентов, протоколирование в БД IBM Lotus. Решение обрабатывает впечатляющий по объему поток сообщений и имеет очень хорошую горизонтальную масштабируемость.

Центральное звено

Для приложений, построенных на базе Websphere MQ, характерны следующие шаблоны использования: маршрутизация, основанная на содержимом сообщения, операции с базой данных, переформатирование контента, работа с адаптерами к информационным системам, публикация/подписка. Все эти шаблоны реализует продукт Websphere Message Broker, с которым мы, выявив все приведенные выше сценарии в процессе разработки и эксплуатации SMSB, начали работать в 2003 году.

Возможны разные подходы к внедрению SOA в организации; по мнению автора, наиболее реалистичный — эволюционный, через накопление сервисов в ESB и последующее внедрение решения по управлению бизнес-процессами, которое позволит включать сервисы в бизнес-процессы, протяженные во времени.

Тогда же руководство ИТ-подразделения банка осознало проблему роста числа межсистемных взаимодействий и пришло к выводу о необходимости приобретения ПО, которое бы стояло над транспортным уровнем Websphere MQ и реализовывало перечисленные выше шаблоны, а также необходимости управления потоками работ (workflow). После проведения пилотных проектов был выбран IBM Websphere BI Server, включавший в себя IBM Websphere Message Broker и IBM Websphere MQ Workflow.

Websphere Message Broker (MB), работающий под управлением Windows, представляет собой сервис операционной системы и именованный объект, ассоциированный с менеджером очередей. Сервис управляет работой экземпляров процесса ОС, и уже в этих процессах запускаются нити ОС, исполняющие потоки обработки сообщений. Кроме того, в системе есть вспомогательные компоненты, такие как Configuration Manager, позволяющий внедрять наши разработки в брокер, и User Name Server для управления правами пользователей на публикацию/подписку в гетерогенной среде. В общем, продукт непростой. И, как говорится в одном анекдоте: «А теперь мы попробуем со всем этим взлететь».

На взлете

Что подразумевает использование брокера обработки сообщений — центрального компонента в нашей архитектуре обмена информацией между приложениями? Продавцы таких продуктов любят показывать две картинки: первая — хаос межсистемных связей; вторая — добавили брокер, замкнули все связи на него, и все стало кристально ясно!

Но наш человек, готовый к подвоху, понимает — просто так сложность никуда деваться не могла. Сложность информационных систем, подобно энтропии, никак уменьшаться не хочет. Она просто спряталась в тот самый центральный прямоугольничек на схеме, в брокер, и, если не принять меры, радикально отравит нам жизнь уже на следующем витке развития.

Коль скоро мы уловили, что сложность, по сути, сходна с энтропией, то и бороться с ней будем упорядочением — названий объектов (очередей, менеджеров сообщений), артефактов разработки (проектов, файлов, функций и т. п.), групп исполнения брокера, потоков обработки сообщений и т. д. Однако стандарта именования объектов недостаточно. Нужно выявить основные шаблоны разработки и воплотить их в отдельных потоках обработки сообщений, обеспечивающих связь с окружающим миром через интерфейсные очереди.

Таким образом, первым этапом в первом проекте с использованием Websphere MB, связанным с системой SMSB, было написание документа по стандартам разработки. Проект выполняла компания VDI, партнер IBM, имеющая опыт создания приложений на Websphere MB. В результате SMSB была переведена на более надежную и масштабируемую платформу, добавлен целый набор услуг. В таком состоянии система находится и поныне, правда, клиентская база увеличилась со времени внедрения более чем в 20 раз. Столь плодотворным в плане начала использования Websphere MB, выработки стандартов, развертывания работающей конфигурации на кластере под управлением ОС Microsoft Windows стал для нас 2004 год.

Выращиваем позвоночник

Получив в свое распоряжение мощный центральный механизм обработки потоков данных, мы начали реализовывать проекты по интеграции систем. Для подключения информационных систем к интеграционной шине было решено использовать механизм адаптеров вместо прямых подключений Websphere MB. Причин этому несколько. Во-первых, мы не хотели нагружать Websphere MB функциональностью, не совсем для него свойственной. Система имеет средства, с помощью которых вполне можно подключиться практически к любой информационной системе (ИС) в банке. Но быстрее всего Websphere MB обрабатывает сообщения Websphere MQ, именно на это направлена его функциональность, и, учитывая, что Websphere MB лицензируется по процессорам и лицензия стоит недешево, имело смысл передать функции взаимодействия с ИС специальным программам-адаптерам.

Другое соображение — гибкость по отношению к топологии сети банка. Представим, что надо взаимодействовать с ИС, которая находится за пределами сетевого экрана (например, шлюз ММБ в нашем случае), или с географически удаленной ИС с ненадежной связью. Такие топологические проблемы прекрасно решает Websphere MQ, к которой ИС подключаются с помощью адаптеров на основе технологии IBM Websphere Business Integration Adapter (WBI) Framework либо посредством адаптеров, разработанных пользователями. Кроме того, готовые адаптеры могут предоставляться производителями ИС, например, мы использовали MQадаптер к системе Kondor+ разработки компании Reuters.

Еще одним доводом в пользу применения адаптеров являются менее жесткие требования по надежности. Допустим, что у какой-либо ИС возникли проблемы. Если эта система связана непосредственно с Websphere MB, то проблемы могут эскалироваться на весь интеграционный брокер или как минимум на группу исполнения. Если же с системой связан адаптер, проблемы хорошо локализуются. Тут, пожалуй, будет уместна аналогия с периферической нервной системой. Представьте себе, что не было бы позвоночника, собирающего и транслирующего нервные импульсы с периферии тела, а вся информация, так же как и зрительная, шла прямо в мозг. Это годится, наверное, для сказочного Колобка, а не для человеческого тела с его универсальностью и красотой.

Мы начали использовать адаптеры для подключения к информационным системам в 2005 году. Для подключения через базу данных был приобретен IBM WBI JDBC Adapter, для подключения к шлюзу ММБ MICEX gateway написан адаптер с использованием C++ API на основе WBI Framework. С системой Kondor+, как уже упоминалось, взаимодействие реализовано через адаптер производителя самой ИС. Выбор в пользу того или иного адаптера должен быть мотивированным. WBI Adapter — стандартизованное решение, его настройка на конкретную ИС, как правило, заключается в редактировании конфигурационных данных. С другой стороны, если производитель ИС предоставляет адаптер, лучше пользоваться им, поскольку есть надежда, что производитель сам в дальнейшем поддержит обратную совместимость интерфейсов ИС. Для систем на AS/400 был разработан собственный Message Adapter, хотя в принципе можно было использовать тот же JDBC Adapter для вызова программ на AS/400 как хранимых процедур. Однако AS/400 представляет собой великолепный сервер приложений с полной поддержкой транзакционности при обработке запросов, входящих по Websphere MQ! Неразумно было этим не воспользоваться.

Еще одним важным и интересным проектом, стартовавшим в 2005 году, стала разработка Единой системы платежного документооборота (ЕСПД) на базе Websphere MB и Websphere MQ Workflow. Назначение ЕСПД — прием платежных документов из каналов дистанционного банковского обслуживания (ДБО) и проведение их по маршруту выполнения, который, как правило, заканчивается в целевой АБС (в настоящее время это может быть как MIDAS DBA, так и FlexCube Core).

Первые сервисы

Для решения задачи повторного использования кода в Websphere MB существуют различные способы: библиотеки функций ESQL, пользовательские узлы, подпотоки (subflows), выделенные потоки обработки сообщений, с которыми можно общаться через интерфейсные очереди. Преимуществами последнего способа являются автономность и масштабируемость, которая легко достигается благодаря использованию дополнительных экземпляров. Кроме того, за интерфейсом можно скрывать детали реализации, которые могут меняться со временем, например переход с одного механизма отправки SMS на другой.

Таким образом, «оборачивать» в такие потоки обработки сообщений имеет смысл автономные задачи, которые потенциально могут быть задействованы несколькими приложениями-клиентами. Поскольку эти компоненты «живут» в интеграционной шине, они в принципе доступны всем информационным системам банка. То есть речь идет о крайне популярном сейчас архитектурном шаблоне — о сервисах. Не хватает «мелочи» — интерфейс должен быть хорошо специфицирован. В данном случае сервисы рассматриваются как архитектурный шаблон, а не как реализация тройки SOAP-WSDL-UDDI, поэтому решение проблемы специфицированности интерфейса было административное, а не техническое.

Говоря о реализации сервисов, трудно обойти вопрос: «А кому, это, собственно, нужно?» Поначалу он вызывает недоумение: «Но позвольте, вы же сами говорили: повторное использование и все такое... То есть сервисы — это хорошо, это нужно, это надо использовать!» Но проблема состоит в том, что кто-то в организации — будь то человек или подразделение — должен выделять сервисы, формализовывать интерфейсы, заставлять разработчиков придерживаться выбранных подходов, а не делать побыстрее и попроще. Мы говорим об организации, в которой объем разработок таков, что один человек не в состоянии контролировать всю разработку, где приходится прибегать к внешним разработчикам и где проекты идут достаточно изолированно один от другого в том смысле, что мало сотрудников участвует сразу в нескольких крупных проектах. Там же, в проектах, изолируются и полученные в ходе выполнения знания, проект также характеризуется тем, что основной интерес его участников — выполнение проектного задания, а не реализация каких-то сервисов. Должны быть люди в организации, заинтересованные не только в текущем удовлетворении запросов бизнеса, но и в получении гармоничной ИТ-структуры. И не только заинтересованные, но и обладающие достаточными полномочиями для того, чтобы определять архитектуру получаемых систем.

Наши первые интеграционные проекты выполнялись в идеологии точка-точка, хотя и с применением интеграционной шины, что только усложняло решение по сравнению с традиционными механизмами. Причиной этого как раз и был явный приоритет интересов конкретных проектов перед интересами ИТ-архитектуры в целом. Ситуация была исправлена, когда в ММБ зимой 2006/2007 года был создан архитектурный отдел.

На оперативном просторе

Естественно, задачи архитектурного отдела гораздо шире, чем выделение сервисов. Тут и прикладная архитектура, и интеграционная архитектура, и архитектура данных. Кроме того, деятельность отдела во многом подчинена выполнению важнейшей задачи, стоящей перед ИТ банка, — внедрения новой АБС FlexCube Core. Данная АБС имеет адаптер FLEXXML, разработанный фирмой — поставщиком АБС. Архитектурный отдел стал практически центром компетенции по использованию FLEXXML и взаимодействию с разработчиками для запуска новых и доведения до ума существующих интерфейсов FLEXXML. Кроме того, в связи с внедрением приходится решать задачи, связанные с перераспределением функциональности по компонентам обеих АБС и описанием возникающих при этом интерфейсов, а также участвовать в разработке интерфейсов или контролировать этот процесс.

На оперативном просторе

Несколько советов по внедрению интеграционной шины IBM Websphere и ведению интеграционных проектов

- По возможности вырабатывайте стандарты.
- В проектах как можно раньше допустите разработчиков к реальным (но, конечно, не производственным) системам. Представления разработчика, сложившиеся на основе документации, могут коренным образом отличаться от реальности.
- Вовремя и тщательно продумайте политику кластеризации Websphere MQ, что позволит в дальнейшем получить горизонтальную масштабируемость.
- Накапливайте повторно используемые компоненты, создавайте сервисы. Не жалеете на это сил и влияния, со временем это окупится.

Один из интересных реализуемых при этом проектов — создание сервиса ABSService2, который позволяет внешним системам не знать о том, что счета могут содержаться в двух АБС. Внутри сервиса есть компоненты, которые маршрутизируют запрос или направляют его в две системы, а потом агрегируют ответы. Продолжение следует...

Сейчас мы пытаемся накапывать сервисы и стремимся создать ситуацию, когда при достаточном количестве сервисов произойдет качественный переход: реализовывать бизнесзадачи проще будет в идеологии SOA. Таким образом, вдумчивое использование интеграционной шины предприятия (ESB) создает предпосылки для SOA в организации.

Возможны разные подходы к внедрению SOA в организации; по мнению автора, наиболее реалистичный — эволюционный, через накопление сервисов в ESB и последующее внедрение решения по управлению бизнес-процессами, которое позволит включать сервисы в бизнес-процессы, протяженные во времени. Оптимизация приказанном, при помощи крупного проекта, видимо, возможно, обойдется очень дешево. Вспоминать SOA кажется путь, когда в рамках небольшого проекта вырабатываются и принимаются концепции, стандарты и т. п. и потом используются в эволюционном переходе к SOA.

