

Система расчета стоимости звонков (Rating System)

Данный пример демонстрирует один из элементов сервис-ориентированного подхода к интеграции приложений с использованием WebSphere Integration Developer. Рассматривается создание гибких и динамичных приложений, хорошо приспособленных к изменяющимся условиям ведения бизнеса и не требующих для этого перестройки всего проекта. Демонстрируется использование **бизнес-правил** - ключевого механизма, позволяющего придавать вышеуказанную гибкость приложениям.

Бизнес-процесс – система или процедура, позволяющая организации достичь цели более высокого уровня. Бизнес-процесс может быть автоматизированным, может представлять собой последовательность шагов, которые необходимо выполнить вручную. Бизнес-процессы могут быть как кратковременными, так и длительными.

Бизнес-процессы часто включают в себя **ключевые бизнес-решения**, отражающие текущую стратегию ведения бизнеса.

В качестве примера можно привести механизм сезонных скидок для особых клиентов компании. Величина скидки является ключевым показателем, зависящим от категории клиента и сезона. Разумеется, компании хотелось бы иметь возможность быстро и просто изменять размеры скидок в соответствии с текущими условиями.

Можно бы было каждый раз при изменении скидок перестраивать приложение, или же поместить информацию о текущих скидках в базу данных, из которой приложение получало бы информацию обо всех изменениях.

Но эти способы слишком накладны. Помимо переустановки приложения и организации взаимодействия с базой данных, оно требует специальных навыков от сотрудника, вносящего изменения. Данный демонстрационный пример показывает, что механизм бизнес-правил позволяет вносить изменения в работу системы быстро, наглядно и без лишних затрат.

Постановка задачи

Рассматривается телефонная компания, в которой тарификация звонков проводится следующим образом:

- Звонок делится на две фазы: первая – фаза ожидания ответа вызываемого абонента; вторая фаза начинается после ответа вызываемого абонента. Каждый из этих этапов оценивается по своему тарифу.
- Стоимость звонка на каждом этапе рассчитывается по формуле:
цена * длительность.
- Длительность каждой фазы измеряется в секундах и округляется в соответствии с тарифом. (Например, если величина округления равна 30 секундам, абонент ожидал ответа 61сек и разговаривал 56 сек, то при расчете стоимости длительность фазы ожидания округлится до 90 сек, фазы разговора – до 60 сек).
- Общая стоимость звонка вычисляется как сумма стоимостей двух этапов
(тариф1*длит.1+тариф2*длит.2).

Тариф и его составляющие (цена минуты разговора, величина округления) могут изменяться в режиме реального времени, внесение изменений не должно быть трудоемким делом.

В данном примере мы покажем, как используются бизнес-правила для определения и модификации тарифов «на лету».

Бизнес-правила

Для реализации бизнес-логики (в нашем случае это правила выбора тарифа и расчета стоимости) определяются соответствующие бизнес-правила.

В среде WebSphere Integration Developer бизнес-правила могут задаваться в виде набора правил (Rule set) или в виде таблицы (Decision table). Правила объединяются в группу (Rule group).

Rule sets

Набор правил можно организовать по принципу if-then. В этом случае определяется набор условий и набор действий, выполняемых при истинности условия. Все условия из этого набора проверяются последовательно, таким образом, в результате может быть выполнено сразу несколько действий.

Можно также определить действия, выполняемые вне зависимости от входных данных (Action rules).

Decision tables

Второй вариант определения бизнес-правил – табличный.

Так же, как и для набора правил if-then, в таблице указываются связи между условиями и выполняемыми действиями. Действие выбирается исходя из истинности нескольких условий. Каждый столбец и каждая строка определяет некоторое условие. Выполняемое действие записывается в клетку на пересечении тех условий, которые предполагаются истинными.

Можно также указать действие, которое будет выполняться в том случае, когда ни одна клетка таблицы не может быть выбрана (otherwise condition).

WebSphere Integration Developer позволяет создавать шаблоны, что значительно упрощает работу, если необходимо завести множество однотипных правил. Правила, построенные с помощью шаблонов, могут быть легко отредактированы.

Правила объединяются в группы. Группа служит для лучшей структуризации бизнес-правил и может содержать критерии, исходя из которых те или иные правила становятся активными.

Business Rules Manager

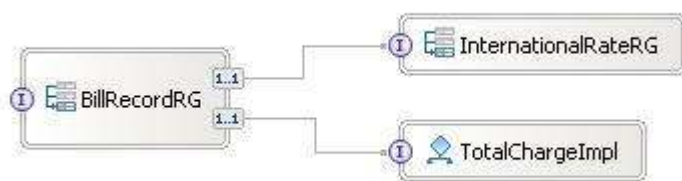
Веб-клиент Business Rules Manager позволяет в режиме реального времени изменять параметры бизнес-правил, созданных на основе шаблонов, а также добавлять новые правила. Все осуществляемые изменения можно документировать.

Те же изменения можно осуществить и в среде разработки, но у Business Rules Manager есть ряд преимуществ:

- простой и наглядный интерфейс, скрывающий все детали реализации, кроме шаблонных бизнес-правил;
- можно посмотреть дату последнего изменения и комментарии к нему;
- изменения вступают в силу только после публикации, в случае ошибки их можно просто не сохранять.

Применение бизнес-правил для решения поставленной задачи.

Строение модуля Rules Manager:



Основную бизнес-логику определяет группа правил BillRecordRG, которая по умолчанию обращается к набору правил getBillingRS.

▼Rule Set

Detailed properties for this rule set.

getBillingRS

▼Interface

	Name	Type
Input(s)	billingRecord	BillingRecord
	billingDate	date
Output(s)	chargeOutput	double

▼Variables

Name	Type
rate1	Rate
rate2	Rate
increment	double

▼Rules

Name	Rule1	
Template	Template_Rule1	
Presentation	If billingRecord.usage_plan == Pay As You Go then increment = 30	
Name	Rule2	
Template	Template_Rule1	
Presentation	If billingRecord.usage_plan == Regular then increment = 6	
Name	Rule3	
Presentation		
Action	rate1=createBO("http://RatingSystem","Rate") rate1.destination = billingRecord.destination1 rate2=createBO("http://RatingSystem","Rate") rate2.destination = billingRecord.destination2	
Name	Rule4	
Presentation		
Action	Partner	InternationalRateRGRef
	Operation	getRate
	Input	input1: rate1
	Output	output1: rate1
Name	Rule5	
Presentation		
Action	Partner	InternationalRateRGRef
	Operation	getRate
	Input	input1: rate2
	Output	output1: rate2
Name	Rule6	
Presentation		
Action	Partner	TotalChargeJavaReference
	Operation	operation1
	Input	record: billingRecord , rate1: rate1 , rate2: rate2 , increment_in_sec: increment
	Output	total_charge: chargeOutput

▼Templates

Name	Template_Rule1		
Presentation	If billingRecord.usage_plan == {param0} then increment = {param1}		
Parameters	Name	Type	Constraint
	param0	string	None
	param1	int	None
IF	billingRecord.usage_plan == param0		
Then	increment = param1		

S&T International

115114, Москва, 2-й Кожевнический переулок, д. 12, стр. 2
Тел.: +7 (495) 255 38 33, факс: +7 (495) 255 38 35

www.sntru.com, info@sntru.com

Компонент `getBillingRS` на вход получает бизнес-объект `BillingRecord`, содержащий все необходимые параметры звонка (`id` клиента, местоположение вызывающего и вызываемого абонентов, продолжительность ожидания и разговора, тарифный план) и возвращает итоговую стоимость звонка.

`getBillingRS` обращается к группе правил `InternationalRateRG`, которая определяет цену в зависимости от территориальной зоны, и к процессу `TotalChargeImpl`, рассчитывающему стоимость отдельных фаз звонка и суммарную стоимость. Полученная стоимость возвращается в качестве параметра `chargeOutput`.

Демонстрация примера

1. Запустите `WS Process Server` (host 10.176.97.11).
2. Запустите `WS Integration Developer` и откройте пример `Rating System`.
3. На вкладке `Servers` выберите `WebSphere Process Server 10.176.97.11` и нажмите кнопку `Start`. Сервер должен придти в состояние `Started`.
4. Теперь можно приступать к непосредственному тестированию примера. Для этого разверните дерево каталога `Rating System`, щелкните правой кнопкой мыши по `Assembly Diagram` и выберите `Test ->Test Module`.
5. На открывшейся вкладке `RatingSystem_Test` проверьте, что выбран компонент `BillRecordRG` и заполните таблицу параметров следующими значениями (обратите внимание на то, что существует различие между строчными и прописными буквами!):

<code>customer_id</code>	1
<code>destination1</code>	US
<code>time_in_sec1</code>	64
<code>destination2</code>	Europe
<code>time_in_sec2</code>	91
<code>usage_plan</code>	Pay As You Go
<code>total_charge</code>	0

billingDate	2002-01-01
-------------	------------

6. Нажмите на Continue. Вы увидите результаты тестирования.
7. Теперь можно отредактировать существующие правила. На вкладке Servers щелкните правой кнопкой мыши по используемому серверу и выберите Launch->Business Rules Manager. Слева вы увидите список книг правил. Откройте BillRecordRG->getBilling->getBillingRS (в нашем примере это единственный набор правил, в котором использованы шаблоны).
8. В окне справа вы увидите все правила, подлежащие редактированию, а также дату внесения последних изменений.
9. Нажмите на кнопку «Изменить».
10. Измените величину округления (increment) в тарифном плане Pay As You Go с 30 на 60.
11. Создайте новое правило. Для этого нажмите на кнопку «Создать правило на основе шаблона». Заполните появившуюся форму:
имя=RulePromo
usage_plan=Promo
increment=1
и нажмите «Добавить»

Режим редактирования: **getBillingRS - Набор правил**

Сообщения:

Общая информация

Последняя публикация	08.окт.2007 12:38 (местное время)	Состояние	*Оригинал
Описание	<input type="text"/>		

Правила

Имя	Правило	Действие
Rule1	If billingRecord.usage_plan == Pay As You Go then increment = 60	↓ Удалить
Rule2	If billingRecord.usage_plan == Regular then increment = 6	↑ Удалить

Шаблоны

Для создания правила заполните поля шаблона и нажмите кнопку "Добавить".

Имя шаблона	Имя	Правило	Действие
Template_Rule1	RulePromo	If billingRecord.usage_plan == Promo then increment = 1	Добавить

12. С помощью стрелок переместите новое правило на верхнюю строчку.
13. Нажмите на кнопку «Сохранить» (в верхней части окна)
14. Изменения сохранены, но еще не вступили в силу. Осталось только опубликовать их.
В списке слева выберите «Опубликовать и вернуться» и нажмите на кнопку «Опубликовать».
15. Теперь протестируйте модуль еще раз (см. п. 4) сначала с прежними параметрами, а потом изменив usage_plan на Promo и убедитесь, что возвращаемые значения поменялись!

Инструкция по разработке данного примера:

http://www.ibm.com/developerworks/websphere/techjournal/0609_gregory/0609_gregory.html