

УДК 004.048:519.816

М.К. Буза, В.С. Горшунов

СИСТЕМА МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ КЛИЕНТОВ ОРГАНИЗАЦИЙ

Исследуются система моделирования поведения клиентов, функционирование и архитектурные решения подобных систем. Предлагается механизм идентификации клиентов, основанный в отличие от известных на заданной целевой функции с использованием контрольных и динамических параметров.

Введение

С каждым годом бизнес становится все более конкурентным, идет объективный процесс его укрупнения. Становится понятно, что бизнес работает для клиента и поэтому должен быть клиентоориентированным. Это включает в себя привлечение новых клиентов, удержание старых и прогнозирование их оттока. Компании нужны ответы на вопросы, какие клиенты способны принести ей наибольшую прибыль, а какие отличаются наибольшей склонностью к смене компании как партнера. Не нарушая общности рассуждений, рассмотрим эти вопросы на примере банковской системы. Основными проблемами банка являются выделение сегмента клиентов, которые с высокой степенью вероятности могут отказаться от услуг банка, установление причин этого отказа и разработка выгодного с точки зрения банка и клиента решения, которое позволит их удержать.

Изучение поведения клиента показывает, что он становится клиентом банка, если банк может помочь решить его проблемы: получить кредит, приобрести мобильный интернет, отправиться в заграничную поездку, поддержать здоровье, получить образование и т. д. Изучать потребности клиента можно различными методами. Наиболее распространенными из них являются сбор и анализ семантических данных, в результате которых информация о всех клиентах обрабатывается и сегментируется. Именно сегментация клиентской базы позволяет увидеть реальную картину текущего состояния клиентов. Однако статистика дает только срез состояния на текущий период, а банку необходимо определить, что будет с клиентами завтра, через год или несколько лет. Десятилетний период кажется дальним горизонтом, поэтому многие компании о нем даже не задумываются. Однако лидеры рынка приходят к выводу о необходимости формирования клиентской базы на будущее. Ниже предлагается один из механизмов идентификации клиентов, основанный на моделировании их поведения [1–3].

1. Модель сегментирования клиентов

Цель модели – ранжирование множества клиентов на основе предложения пакета услуг с максимальной доходностью для банка. Сегментирование предложений банка зависит от принципа взаимодействия с клиентом. Например, клиент сам пришел в банк (reactive) или банк сделает ему предложение. Другим примером сегментирования может служить источник информирования клиента о предложении, которым может быть, например, партнер банка, либо рекламная sms-рассылка, либо сайт банка и т. д. [4].

Предположим, что банк разработал два пакета предложений:

1. Увеличение кредитного лимита клиента (Increase Credit Limit):
 - а) по инициативе клиента (Reactive: Increase Credit Limit);
 - б) по инициативе банка (Proactive: Increase Credit Limit).
2. Уменьшение годовой процентной ставки по кредиту (Reduce Annual Percentage Rate – Reduce APR):
 - а) по инициативе клиента (Reactive: Reduce APR);
 - б) по инициативе банка (Proactive: Reduce APR).

В модели предложения банка присутствуют как постоянные (контрольные) параметры, так и динамические.

Постоянные параметры включают:

1. *Пороговую благосклонность* клиента (Propensity Threshold) – минимальную возможную благосклонность клиента, при которой банк будет считать целесообразным делать данное предложение. Значения этого параметра изменяются в интервале от 0 до 1. В предложенной модели значения определены следующим образом:

0 – клиент не склонен рассматривать предложение банка;

0,25 – скорее нет, чем да;

0,5 – клиент размышляет над предложением;

0,75 – скорее да, чем нет;

1 – клиент готов принять предложение банка.

2. *Маркетинговый вес* (Marketing Weight) – показывает ценность данного предложения в стратегическом плане для банка. В рассматриваемой модели параметр принимает дискретные значения от 0 до 1 с шагом 0,25. Семантика значений предложений следующая:

0 – наименьшей ценности;

0,25 – малой ценности;

0,5 – средней ценности;

0,75 – высокой ценности;

1 – наибольшей ценности.

3. *Уровень сегментации* (Champion Challenger) – определяет пороговое значение (Champion), при превышении которого значением-кандидатом (Challenger) предложение может быть выставлено клиенту. Значения параметра изменяются в интервале от 0 до 1.

4. *Стимулирование клиента* (Incentive) – показывает, будет ли выполнено стимулирование клиента в контексте текущего предложения. Принимает значения:

0 – банк не будет стимулировать клиента;

1 – банк будет стимулировать клиента.

5. *Время действия предложения* (Valid To) – предельное время, до которого будет действовать предложение банка.

Динамические переменные параметры модели включают:

1. *Склонность клиента* принять предложение банка (Propensity) – интервал изменения от 0 до 1. Принимает значения, аналогичные значениям параметра пороговой склонности клиента.

2. *Доход* (Revenue) – вычисляется как сумма кредита, умноженная на процент кредита.

3. *Применимость предложения* (Fulfilled) – показывает, было ли принято решение о выдаче предложения банка клиенту. Принимает значения: 0 – не применимо, 1 – применимо.

Проанализируем структуру данных, состоящую из контрольных и динамических параметров модели предложения. Применительно к данному клиенту будем рассматривать два различных варианта этой структуры:

– *текущее состояние* (Champion), при котором для клиента рассматривается возможность выдачи предложения;

– *последующее состояние* (Challenger), при котором клиент принял предложение.

Принципы заполнения данных структур: контрольные параметры модели задаются маркетологами банка (пользователями модели); переменные параметры модели выбираются из базы накопленных данных, учитывающих предысторию работы с клиентами.

Из всех предложений, предлагаемых клиенту, выбирается то, у которого расчетное значение *веса* *веса* *коэффициента* (weighting factor – wf) вычисляется по формуле

$$wf = \begin{cases} 0 & \text{if propensity} < \{\text{Propensity Threshold}\}; \\ x_i & \text{if propensity} \geq \{\text{Propensity Threshold}\}, \end{cases}$$

где $x_i = \text{propensity} * \{\text{Marketing Weight}\}$.

В зависимости от весового коэффициента выбираем предложение, попадающее в интервал его значений:

```

if 0 <= wf <= 0,25 then Churn.Reactive.Reactive: Increase Credit Limit;
if 0,25 < wf <= 0,5 then Churn.Reactive.Reactive: Reduce APR;
if 0,5 < wf = 0,75 then Churn.Proactive.Proactive: Increase Credit Limit;
if 0,75 < wf <= 1 then Churn.Proactive.Proactive: Reduce APR.

```

После этого происходит обработка полученных результатов. Структуру, содержащую приведенные данные, назовем фактом (Fact).

В обработке участвуют следующие данные:

- 1) индекс предложения (Proposition Id) – уникально идентифицирует предложение, например Proactive: Increase Credit Limit;
- 2) источник информации клиента о предложении (Channel);
- 3) параметр доходности (Revenue) – берется из соответствующего поля модели предложения;
- 4) параметр применимости (Fulfilled) – берется из соответствующего поля модели предложения;
- 5) дата выполнения предложения.

Предложенная модель чувствительна к количеству используемых параметров: с увеличением числа параметров время моделирования увеличивается.

Целесообразно представлять модель на подмножестве языка XML [5, 6]. Это подмножество поддерживает работу как с примитивными типами (число, строка, булево значение, дата и т. д.), так и с массивами этих примитивных типов. Модель включает правила, которые связывают атрибуты клиентов (входные параметры). На выходе ее получаем реальные предложения с вероятностью принятия клиентом предложения банка.

2. Симуляция

Рассмотрим схему работы модели (рис. 1). На вход модели поступает информация о клиенте, значения контрольных параметров (Control Parameters values) и список предложений (Proposal list), на выходе из модели получаем предложение для клиента, которое в случае его принятия записывается в базу накопленной статистики как факт.

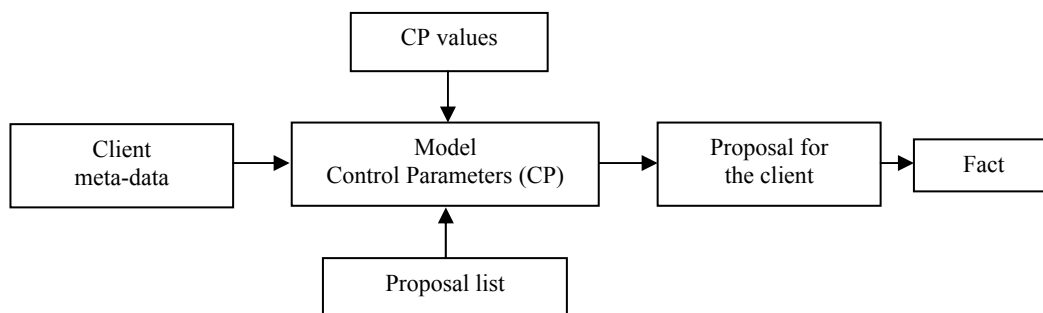


Рис. 1. Схема работы модели

На основании работы модели конкретному клиенту подбирается и делается конкретное предложение, которое он может принять или отклонить. Данные о каждом клиенте попадают в базу накопленной статистики и ранжируются по приведенным выше параметрам. Таким образом, формируется набор исторически накопленных фактов, который позволяет перейти к этапу прогнозирования доходности банка в зависимости от клиентуры банка и предложений клиентам. Этот процесс назовем процессом симуляции (Simulation). Он выполняется на основе исторически накопленных фактов с помощью генетических алгоритмов, которые строят предикативные значения показателей.

Перед началом работы модели произведем генерацию исторически накопленных фактов. Процедура генерации фактов называется *разогревом системы* (Warming up). Вместо генерации можно использовать реальные данные. Предположим, что имеется 10 000 исторически накопленных фактов.

Рассмотрим пример симуляции нашей модели. До начала симуляции модель имеет контрольные параметры (рис 2), которые характеризуют текущее поведение модели. Симулированные данные на их основе будем называть целевыми (target).

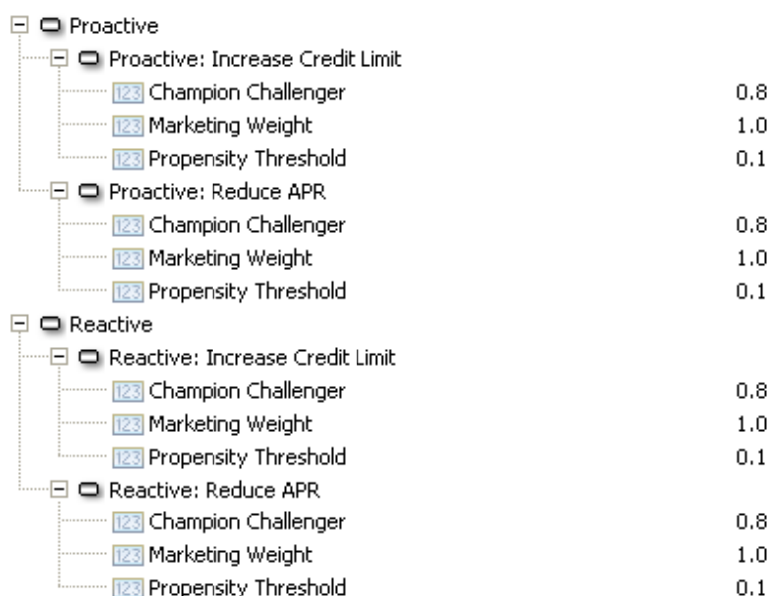


Рис. 2. Начальные значения контрольных параметров модели

Изменим маркетинговый вес для двух контрольных параметров (табл. 1).

Таблица 1

Изменения в маркетинговом весе проактивных предложений

Предложение	Начальное значение	Новое значение
Proactive: Increase Credit Limit	0,0	0,25
Proactive: Reduce APR	0,0	0,5

Выполним процесс симуляции. Полученные в результате этого процесса данные будем называть симулированными (simulated). По его завершении строится таблица изменений в объеме принятых предложений (симулированных результатов) по сравнению с целевыми результатами (табл. 2).

Таблица 2

Изменения в объеме принятых предложений

Предложение	Consumer: New	Consumer: Mature	Consumer: Developing
Reactive: Reduce APR	0	0	0
Reactive: Increase credit Limit	0	0	0
Proactive: Reduce APR	340	400	390
Proactive: Increase Credit Limit	210	140	70

Из табл. 2 виден рост принятых предложений клиентами при условии изменения контрольных параметров по сравнению с начальными параметрами модели. Данные в таблице по-

казывают изменения в количестве принятых предложений клиентами при условии изменения маркетингового веса по сравнению с прогнозируемым уровнем принятия предложений.

Проведем еще одну симуляцию, но уже с изменением маркетингового веса реактивного предложения (Reactive: Reduce APR) с начального значения 0,0 на новое значение, равное 0,75.

В результате получим следующие изменения в объеме принятых предложений по сравнению с прогнозируемым (табл. 3).

Таблица 3

Изменения в объеме принятых предложений			
Предложение	Consumer: New	Consumer: Mature	Consumer: Developing
Reactive: Reduce APR	40	120	60
Reactive: Increase credit Limit	60	10	90
Proactive: Reduce APR	340	400	390
Proactive: Increase Credit Limit	210	140	70

Для визуализации полученных результатов используются метрики, вычисляемые на основе целевых и симулированных фактов. Метрики могут быть предопределенными и пользовательскими (рис. 3). Метрика задается именем метрики (Name), ее типом (Type), который может принимать значения «текущий объем» (Volume) или «кумулятивный объем» (Cumulative), а также формулой для расчета метрики (Expression).

The screenshot displays a 'Metrics' configuration window. At the top, there is a table with columns 'Name', 'Type', and 'Expression'. Below the table are 'Add ...' and 'Remove' buttons. The 'Metric details' section includes fields for 'Name*' (Metric1), 'Expression*' (2*{Fullfillments Processed}), 'Type' (Volume), 'Target comparison' (Normal selected), and an 'Enabled' checkbox.

Name	Type	Expression
Volume	Volume	Volume
Number Accepted	Volume	Number Accepted
Responses processed	Volume	Responses processed
Revenue	Volume	Revenue
Number Fulfilled	Volume	Number Fulfilled
Fullfillments Processed	Volume	Fullfillments Processed
Metric	Volume	Revenue+23
Metric1	Volume	2*{Fullfillments Processed}
Metric2	Volume	1+2+Revenue
Metric3	Volume	1

Metric details

Name*: Metric1 Description

Expression*: 2*{Fullfillments Processed} X+ =0

Type: Volume

Target comparison: Normal Reversed

Enabled

Рис. 3. Конфигурация метрик

В данной модели предопределены следующие метрики:

- общий объем фактов (Volume);
- количество принятых предложений банка клиентами (Number Accepted);
- количество обработанных ответов от клиентов (Responses processed);
- доход (Revenue);
- общее количество выполняемых предложений (Number Fulfilled);
- количество обработанных выполняемых предложений (Fullfillments Processed).

3. Реализация системы сегментирования клиентов

Согласно стандарту IEEE 1472000 «Порядок описания архитектурных решений программных систем, рекомендуемый IEEE» [7] под архитектурой понимается фундаментальное устройство системы, воплощенное в ее компонентах и связях между ними, а также среды и руководящих принципов их дизайна и эволюции.

Отсюда следует, что архитектура программы представляет собой структуру, которая включает элементы программного обеспечения, внешне видимые свойства этих элементов и отношения между ними.

Для работы рассматриваемой модели будем применять вычислительную сеть (грид). На каждом узле этой сети используется база данных Berkeley DB для хранения и обработки исторических и симулированных фактов. Для ускорения выполнения запросов результатов работы с вычислительных узлов применяется кеширование на управляющем узле. Система поддерживает работу с несколькими клиентами одновременно, а также построение нескольких визуализаций результатов как для одной, так и для множества метрик. Более того, пользователи системы могут сохранять выбранные ими представления данных для дальнейшего анализа.

В рассматриваемом случае архитектура системы состоит из следующих основных компонентов (рис. 4):

- вычислительного сервера (App Server);
- управляющего грид-узла (Master Grid Node);
- вычислительных грид-узлов (Grid Nodes);
- хранилища данных (Data store);
- пользовательской системы (User).

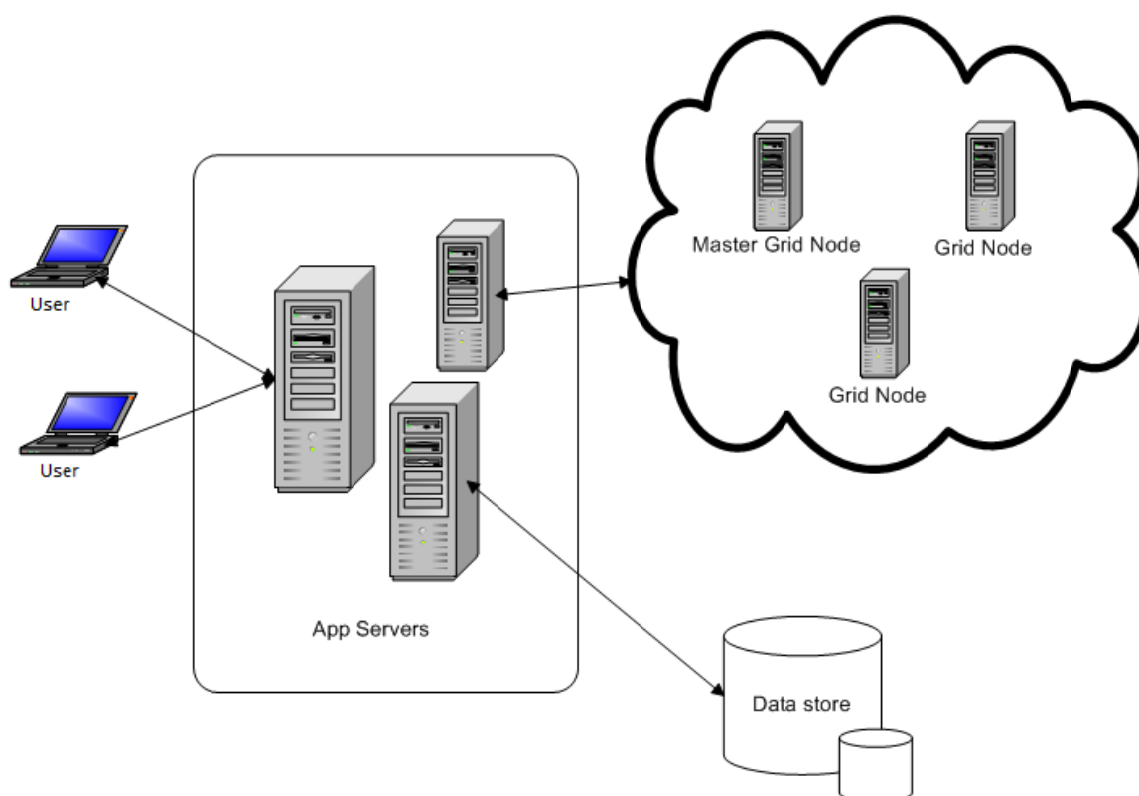


Рис. 4. Архитектура системы сегментирования клиентов

Вычислительный сервер выполняет задачу взаимодействия пользователей и вычислительной сети, а также кеширования данных.

Хранилище данных содержит исторические данные и предоставляет к ним доступ.

Управляющий узел инициирует процесс симуляции, распределяет исторические данные по вычислительным узлам, собирает промежуточные результаты, а также статистику выполнения симуляции с грид-узлов.

На грид-узлах непосредственно выполняется процесс. Пользователь системы является приложением, обеспечивающим возможность работы с параметрами модели, запуска и контроля выполнения симуляции, а также визуализации ее результатов.

Рассмотрим этапы поведения системы в процессе реализации модели (рис. 5). До запуска процесса управляющий узел распространяет на вычислительные узлы документ описания модели на языке XML. Узлы выстраиваются по топологии «кольцо», при этом происходит присвоение каждому узлу уникального ключа, который в процессе симуляции будет использован для идентификации данных, получаемых из вычислительных узлов.



Рис. 5. Процесс реализации модели

Весь процесс симуляции условно можно разбить на четыре этапа. На первом этапе происходит распределение фрагментов данных, участвующих в симуляции между вычислительными узлами. На втором этапе происходит запуск процесса моделирования поведения для получения результатов без изменения исходных параметров. На третьем этапе запускается поиск решений для определения значений фактов с измененными контрольными параметрами системы. Все найденные варианты хранятся в локальной базе вычислительного узла. На последнем этапе управляющий узел запрашивает у каждого узла агрегированные результаты, рассчитываемые по метрикам, которые задаются пользователем модели.

Данные, хранимые на вычислительном узле, описываются двумя структурами – DecisionData и DataFact. DecisionData содержит в себе конкретные значения входных параметров модели, а также возможные значения ее контрольных параметров. Одна запись DecisionData уникальна во всей системе и идентифицируется соответствующим образом. DataFact содержит набор из симулированного и целевого фактов.

В связи с тем что такие наборы данных не зависят друг от друга, их можно обрабатывать параллельно. Благодаря этому достигается атомарность шагов симуляции, в силу чего данный процесс попадает в класс хорошо распараллеливаемых задач.

В основе процесса симуляции лежат генетические алгоритмы. Кроме того, количество целевых и симулируемых фактов прямо пропорционально количеству исторически накопленных фактов и контрольным параметрам системы. В связи с этим для ускорения процесса моделирования целесообразно применить вычислительную сеть, описанную ранее. Атомарность шагов симуляции позволяет получать результаты уже в процессе симуляции, что, в свою очередь, дает возможность пользователю модели анализировать результаты, не дожидаясь окончания процесса. Более того, атомарность операций позволяет вносить изменения в наборы контрольных параметров, не прерывая симуляцию. Новые факты в таком случае будут учтены только на шагах, еще не обработанных к моменту их наступления.

Заключение

Разработанная и предложенная система моделирования поведения клиентов реализована на языке Java 6 применением библиотек параллельного вычисления GridGain. Управляющий сервер располагается на наборе J2EE контейнеров, для хранения информации поддерживается ряд корпоративных баз данных.

На основании сегментации клиентской базы система позволяет определить наиболее прибыльный сегмент клиентов не только в краткосрочной, но и в долгосрочной перспективе.

В отличие от существующих аналогов, таких как решения компаний Oracle и PegaSystems, разработанная система позволяет не только прогнозировать текущее поведение клиентов, но и симулировать поведение с измененными контрольными параметрами. Новизна работы заключается в реализации данной системы с применением алгоритмов симуляции, а также вычислительной сети.

В общем случае решения, следующие из результатов работы предложенной модели, полезны как для оперативного прогнозирования, так и для принятия обоснованных решений в организациях, не преследующих коммерческих целей, в торговых или банковских системах [8].

Список литературы

1. GridGain 3.0: High Performance Cloud Computing [Electronic resource]. – Mode of access : http://www.gridgain.com/media/gridgain_white_paper.pdf. – Date of access : 7.06.2012.
2. Brandic, I. High-level Composition of QoS-aware Grid Workflows: An Approach that Considers Location Affinity / I. Brandic, S. Pllana, S. Benkner [Electronic resource]. – Mode of access : http://www.isi.edu/works06/brandic_works06.pdf. – Date of access : 7.06.2012.
3. Customer Centricity in Insurance Sales, Service and Saving e Customer [Electronic resource]. – Mode of access : <http://www.insurancetech.com/whitepaper/Customer-Insight-Business-Intelligence/Analytics/customer-centricity-in-insurance-sales-service-an-wp1330111991>. – Date of access : 7.06.2012.
4. Peppers, D. Return on Customer: Creating Maximum Value From Your Scarcest Resource / D. Peppers, M. Rogers. – USA : Doubleday, division of random House Inc., 2005. – 304 p.
5. Bass, L. Software Architecture in Practice / L. Bass, P. Clements, R. Kazman. – Second Edition. – USA : Addison Wesley, 2003. – 560 p.
6. Smith S. Managing the Customer Experience: Turning customers into advocates / S. Smith, J. Wheeler. – USA : Financial Times Press, 2002. – 272 p.
7. IEEE Computer Society, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems: IEEE Std 1472000, 2000.
8. Mattison R. Data Warehousing and Data Mining for Telecommunications / R. Mattison. – Norwood, MA : Artech computer Scion Library, 1997. – 237 p.

Поступила 07.06.2012

*Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: bouza@bsu.by, gorshunov@gmail.com*

M.K. Bouza, V.S. Gorshunov

MODELING ORGANIZATION CLIENTS BEHAVIOR

A system of modeling organization clients behavior is proposed and studied, as well as its functional and architectural solutions. A client identification mechanism based on control and dynamic parameters of a given objective function is proposed.