

УДК 004

ББК 32.973стд1-018.2

ПРИЕМ КОММУНАЛЬНЫХ ПЛАТЕЖЕЙ КАК СИСТЕМА МАССОВОГО ОБСЛУЖИВАНИЯ

Клименко А. А.¹

(Южный Федеральный Университет)

В работе рассмотрена система приема коммунальных платежей как система массового обслуживания. Рассмотрена функциональная модель системы в качестве диаграмм прецедентов UML. Разработан паттерн проектирования системы приема коммунальных платежей в нотации BPMN.

Ключевые слова: система массового обслуживания, проектирование бизнес-процессов в нотации BPMN, функциональные требования в Use Case диаграммах UML, паттерны проектирования системы приема коммунальных платежей

Рассмотрим процесс приема коммунальных платежей от населения как систему массового обслуживания. Целью работы является применение нотации BPMN (Business Process Modeling Notation) для описания системы массового обслуживания и разработки шаблона (паттерна) для такого класса задач. Систе-

¹ *Клименко Антон Александрович, магистр прикладной математики и информатики (344090, Ростов-на-Дону, пр. Стачки 182, кв. 120, +79064228388, antklim@gmail.com)*

мы массового обслуживания делятся на три типа, в зависимости от наличия возможности ожидания поступающими требованиями начала обслуживания – системы с потерями, системы с бесконечной очередью и системы с конечной очередью. Систему приема коммунальных платежей можно отнести к системе массового обслуживания третьего типа – система с конечной очередью. Поток требований в этой системе является поток клиентов, которым необходимо оплатить услуги. Обслуживаемым объектом является система учета принятых платежей. Рассматриваемая в работе система включает в себя систему обработки и хранения информации о принятых платежах и контрольно-кассовую машину (ККМ). Системы массового обслуживания можно рассмотреть с точки зрения бизнес-процессов. Система подчиняется набору правил и действует в рамках определенного процесса. В работе рассматривается процесс непосредственного приема платежа, основными действующими лицами которого являются клиент и кассир. Этот процесс является частью процесса приема коммунальных платежей, который включает в себя: прием платежей, первичную обработку данных о платежах (происходит в момент закрытия кассиром рабочей смены), обработка платежей и взаиморасчеты с поставщиком услуг.

На сегодня существует несколько классов методологий описания бизнес процессов:

- Диаграммные модели. К этому классу относятся модели в виде IDEF диаграмм и т.д.
- Математические модели. Примером такого класса моделей являются Сети Петри.
- Языки моделирования бизнес-процессов. Представителем такого класса моделей является нотация BPMN (Business Process Modeling Notation).

В работе использован третий тип модели, так как система приема коммунальных платежей рассматривается как бизнес-процесс. Для описания бизнес-процесса использована нотация

ВРМN, которая позволяет в наглядном виде представить бизнес-процесс.

Первая работа о системах управления бизнес процессами (Business Process Management Systems, ВРMS) была опубликована в 2000 году Ismael Chang Ghalimi. В 2001 году увидел свет первый релиз первой ВРMS. Стандарт нотации Business Process Modeling Notation (ВРMН) был разработан Business Process Management Initiative (ВРMI).

Впервые спецификация ВРЕL (Business Process Management) была опубликована 2002, а уже в мае 2004 была представлена спецификация ВРMН 1.0. Эта спецификация отразила более чем двухгодичные усилия ВРMI Notation Working Group. Основной целью усилий над ВРMН было разработка нотации, понятной всем бизнес-пользователям, начиная с бизнес-аналитика, который создает первичное представление процесса, разработчиков, ответственных за применение технологий, которые будут поддерживать эти процессы, заканчивая бизнес-пользователями, которые будут наблюдать и управлять этими процессами. ВРMН также будет поддерживать внутреннюю модель, которая позволит создавать выполняемый ВРЕL4WS (Business Process Execution Language for Web Service – язык выполнения бизнес-процессов для вэб-сервисов). В 2005 году стандарт был передан ОМG¹ и в 2006 вышел первый релиз ВРMН от ОМG [2].

Сегодня ВРMН поддерживают практически все средства моделирования ведущих поставщиков. Нотация имеет широкий набор графических объектов, которые позволяют описать любой бизнес-процесс. ВРMН также допускает использование паттер-

¹ ОМG – Object Management Group – консорциум, занимающийся разработкой и продвижением объектно-ориентированных технологий и стандартов.

нов¹, которые могут быть взяты за основу при построении моделей других похожих бизнес-процессов.

В рамках общей модели BPMN существует три типа подмоделей:

- **Private** (частные или внутренние процессы) – внутренние процессы определенной организации, обычно workflow (поток работ) или деловые процессы. Например, «Принять заказ» → «Подготовить заказ» → «Получить оплату» → «Доставить заказ»

- **Abstract** (абстрактные или открытые процессы) – представляют собой взаимодействие между частным бизнес процессом и другим процессом или участником. Абстрактными считаются процессы, действия которых имеют связи за пределами частного бизнес-процесса.

- **Collaboration** (совместные или глобальные процессы) – взаимодействие между двумя и более участниками, путем обмена сообщениями. Как правило, это два и более абстрактных процесса на одной диаграмме.

В BPMN нет деления диаграмм на типы «Как есть», «Как должно быть», «С точки зрения». Нотация предоставляет проектировщику самому определять, к какому типу относится та или иная диаграмма.

Однако из практических соображений диаграммы делятся на три уровня:

- **Business Layer** (бизнес уровень) – подразумевает рассмотрение процесса с точки зрения бизнес-аналитика, общее

¹ *Pattern – это «шаблон узнавания», некий обобщенный «рисунок проблемы» (pattern recognition – распознавание образов). Анализируя бизнес-процесс можно найти сходство с рассматриваемыми ранее другими процессами и применить описание ранее рассмотренного процесса к рассматриваемому процессу.*

представление различных бизнес-шагов и оркеструющий их поток (поток определяющий взаимодействие и правила связи между процессами)

- **Functional Layer** (функциональный уровень) – рассматривает процесс с точки зрения ИТ-системы. На этом уровне необходимо знать, что процессы взаимодействуют с базами данных или ERP-системами, но не требуется упоминать специфические API или определенные SQL запросы. Проектируется аналитиком, имеющим опыт работы с ИТ-системами. (Формат обмена сообщениями XML Schema проектируется также на этом этапе).

- **Implementation Layer** (уровень реализации) – фактически договор по реализации деталей процесса. Реализуется как уровень сервиса и процесса реализации, открытие WSDL¹ для других процессов. Проектируется ИТ-инженерами, имеющими четкие знания о приложениях взаимодействующих с процессом. Здесь появляются исключения, компенсации, события таймера [2].

BPMN определяет диаграмму бизнес-процессов (Business Process Diagram (BPD)), которая основывается на технике построения блок-схем, предназначенных для создания графических моделей операций бизнес-процессов. Модель бизнес процессов (Business Process Model) представляет собой сеть графических объектов, которые являются действиями (activities) и управление потоком этих действий, который определяет порядок выполнения действий.

BPD состоит из набора графических элементов. Эти элементы позволяют легко разрабатывать простые диаграммы, которые будут понятны большинству бизнес-аналитиков (диаграммы в виде блок-схем). Элементы подобраны таким образом, чтобы их можно было четко отличить друг от друга, и используют формы, которые привычны большинству проектировщи-

¹ WSDL (англ. *Web Services Description Language*) – язык описания веб-сервисов, основанный на языке XML

ков. Например, деятельности (activities) – это прямоугольники (rectangles), а условия принятия решений – ромбы (diamonds). Необходимо подчеркнуть, что одна из причин дальнейшего развития BPMN – это создание простого механизма для разработки модели бизнес-процессов, которая будет способна поддерживать сложные связи внутри бизнес-процессов. Чтобы осуществить поддержку этих двух противоречивых требований, графические аспекты нотации были организованы в специфические категории [5]. В BPMN можно выделить пять ключевых типов объектов:

- **Activities** (деятельности) – работа которая выполняется участником процесса. Есть два основных вида деятельности – задача и подпроцесс. Задачи могут объединяться в подпроцессы, подпроцессы могут становиться циклами. Например «Заклучить договор», «Выдать товар».

- **Pools, Swimlanes** (области, дорожки) – предназначены для организации деятельностей в диаграммах процесса по участникам процесса. Например, область «ИТ-отдел», «Бухгалтерия».

- **Events** (события) – события, которые влияют на поток процесса. Они могут быть причинами, триггерами¹ или результатами. Существует три типа событий – начальное, среднее и конечное.

- **Gateways** (шлюзы) – представляют точки принятия решений в диаграмме процесса, в результате которых процесс может быть продолжен по одному или более путям. Шлюзы могут определять направление потока на основе данных процесса (data-based) или на основе наступления событий (event-based). Существует три вида шлюзов:

- **Exclusively** (эксклюзивно) – может выполняться только одна из ветвей процесса;

¹ *Триггер – механизм (процедура), который срабатывает при наступлении определенного события.*

- **Inclusively** (инклюзивно) – могут выполняться одна или более ветвей процесса;
- **In Parallel** (параллельно) – все ветви процесса выполняются параллельно;
- **Flow Connectors** (соединители потоков) – соединяют один-к-одному деятельности и шлюзы. Они могут представлять простой поток процесса, ссылку посредством сообщения, условный поток, корректирующий поток, поток исключения.
- **Artifacts** (артефакты) – документы и комментарии.

Прежде чем начать составление BPMN диаграммы процесса приема коммунальных платежей кассиром, необходимо выделить основные элементы модели и связи между этими моделями. Для такого анализа целесообразно использовать еще один тип диаграмм в нотации UML – Use Case Diagram (диаграммы прецедентов). Use Case диаграмма показывает связи между действующими лицами и набором вариантов использования системы или подсистемы, а также связи между этими элементами. Связями являются ассоциации между действующими лицами и вариантами использования и отношения наследования между действующими лицами. Некоторые варианты использования могут быть заключены в ограничивающий прямоугольник, для того чтобы показать их принадлежность к одной подсистеме. Основные визуальные элементы, используемые при построении Use Case диаграмм:

- **Actor** (действующее лицо) – определяет логически связанный набор ролей, которые пользователь некоторой сущности может играть по отношению к этой сущности. Действующее лицо играет определенную роль по отношению к каждому отдельному варианту использования, с которым он взаимодействует. Действующим лицом может быть как человек, так и некоторая другая система, взаимодействующая с проектируемой системой. На рис.1 представлено графическое изображение элемента Actor:



Рис.1 Actor

▪ **Use Case** (вариант использования) – некоторый классификатор, представляющий логическую единицу функциональности проектируемой системы или подсистемы, о которой объявляется в наименовании варианта использования. На рис.2 представлено графическое изображение элемента Use Case:



Рис.2 UseCase

▪ **Communicates** (соединения) – визуальный элемент, используемый для обозначения ассоциации между действующим лицом и вариантом использования. Это прямая линия, связывающая действующее лицо и вариант использования. При необходимости над соединением можно указать множественность связи, то есть определить, будет ли это связь один к одному, многие к одному и т.д. На рис.3 приведен пример связи между актером и вариантом использования.

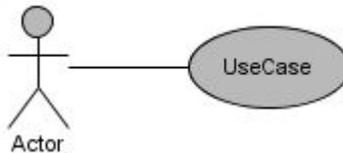


Рис.3 Communicate

▪ **Extends** (расширения)

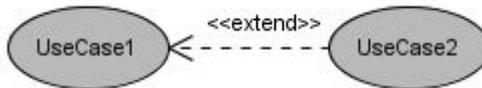


Рис.4 Extends

Визуальный элемент Extends показывает, что при некоторых условиях вариант использования UseCase2 может быть

добавлен к варианту использования UseCase1. При этом UseCase1 является точкой расширения для UseCase2. Расширение изображается стрелкой, направленной от варианта использования, обеспечивающего расширение, к основному варианту использования. Стрелка снабжается пометкой “extend”.

▪ **Includes** (включения)

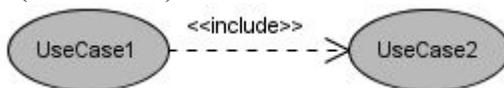


Рис.5 Include

Связь типа include от варианта использования UseCase1 к варианту использования UseCase2 показывает, что вариант использования UseCase1 включает в себя также и поведение специфицированное для варианта использования UseCase2. Включение изображается стрелкой, направленной от основного варианта использования к включаемому варианту использования. Стрелка снабжается пометкой “include”.

Вместо того, чтобы описывать прецеденты непосредственно, лучше использовать сценарий. Сценарий – последовательность шагов, описывающих взаимодействие пользователя и системы. Опишем сценарий «Прием коммунальных платежей» [3].

Плательщик коммунальных услуг представляет кассиру квитанцию. Кассир вводит данные с полученной квитанции (вводит данные плательщика и определяет получателя). Кассир называет плательщику сумму оплаты и принимает деньги. Кассир проводит платеж в системе и отдает плательщику чек.

Подобный сценарий описывает только одну ситуацию, которая может иметь место. Если во время проведения платежа произойдет какой-либо сбой связи, то подобная ситуация может послужить предметом уже другого сценария. В другом случае кассир может обслуживать постоянного клиента, информация о котором уже есть в системе, и вводить информацию по нему не требуется, и это порождает еще один сценарий.

В любом случае, все эти сценарии имеют одну общую цель: у плательщика – оплатить услуги, а у кассира – обслужить плательщика и осуществить платеж. Правда, кассир не всегда может достичь указанную цель. Именно цель является ключом к прецедентам. Сформулируем сценарий прецедента «Прием коммунальных платежей».

Таблица 1. Сценарий процесса приема коммунальных платежей

<p>Прием коммунальных платежей</p> <p>Целевой уровень: уровень моря¹</p> <p>Главный успешный сценарий</p> <ol style="list-style-type: none">1. Плательщик предоставляет квитанцию2. Кассир вводит данные платежа3. Плательщик оплачивает квитанцию4. Кассир проводит платеж и выдает плательщику документ(ы) подтверждающий платеж5. Плательщик получает документ(ы) подтверждающий платеж
--

¹ *Схема уровней прецедентов была предложена в книге Кокборна [4]. На «уровне моря» находится Базовый прецедент. Эти прецеденты обычно представляют отдельное взаимодействие ведущего актера и системы, предоставляют ведущему актеру какой-либо полезный результат. Ведущий актер – это актер, желание которого пытается удовлетворить прецедент и который обычно, но не всегда является инициатором прецедента.*

Расширения:

2а. Плательщик является плательщиком с историей

.1: Система предоставляет информацию о платежах плательщика

.2: Кассир вводит только сумму платежа и переходит к шагу 4 главного успешного сценария

4а. Возникла системная ошибка при проведении платежа

.1: Кассир может либо повторно попытаться провести платеж, либо отменить платеж

Описав сценарий выполнения процесса приема коммунальных платежей, можно перейти к его графическому описанию. Из сценария сразу можно выделить двух актеров – кассир и плательщик.



Рис.6 Кассир и плательщик

На рис.6 изображены основные действия, которые выполняются при приеме коммунальных платежей. Как можно

заметить, прецедент «Получить чек» зависит от прецедента «Напечатать чек». Эту зависимость можно отразить на диаграмме (рис. 7).



Рис. 7 Кассир и плательщик

Полученная диаграмма полностью описывает взаимодействие двух основных актеров – кассира и плательщика. Однако при более глубоком рассмотрении процесса приема коммунальных платежей, можно выделить еще двух актеров:

- ККМ – контрольно-кассовая машина, при помощи которой кассир печатает чек, означающий проведение платежа, который далее передается плательщику
- Система хранения и обработки информации о проведенных платежах (БДП – база данных платежей)

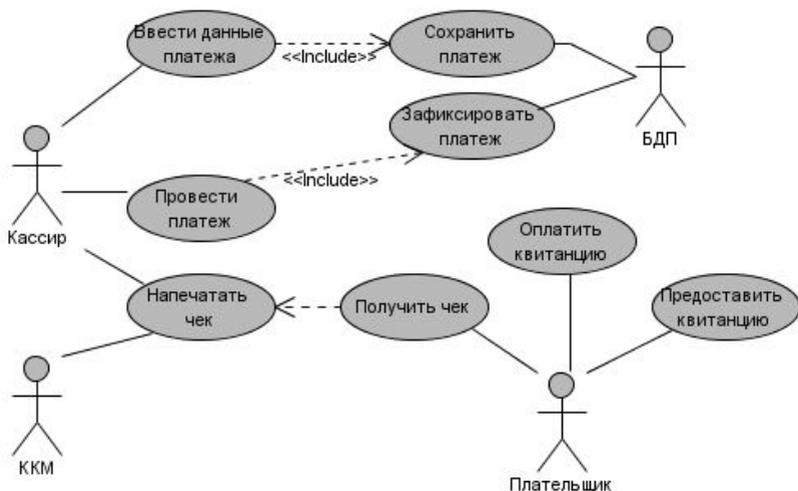


Рис.8 Развернутая система

На рис.8 представлено более полное описание функциональных связей в системе приема коммунальных платежей. После описания прецедентов системы можно перейти к описанию бизнес-процессов в нотации BPMN. Правильно построенная диаграмма прецедентов позволит быстро перейти к рассмотрению бизнес процессов, так как можно установить связь между актерами в диаграмме прецедентов и пулами (дорожками) в диаграммах в нотации BPMN, а прецеденты раскрываются при помощи деятельности. Однако перед тем как описывать процесс приема коммунальных платежей в нотации BPMN, необходимо четко определить, с точки зрения какой деятельности описывается этот процесс. От этого может зависеть правильность построения всей системы в целом. Построим на примере рассматриваемой системы две диаграммы, которые описывают процесс приема платежей, но с точки зрения разных актеров.

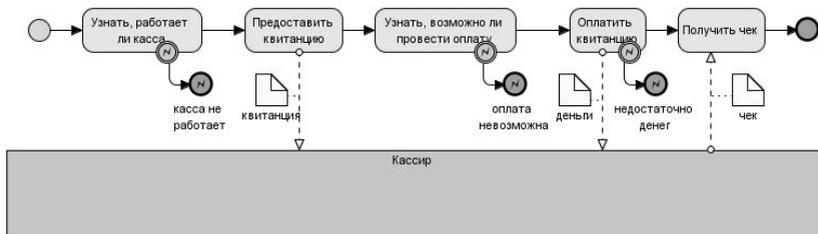
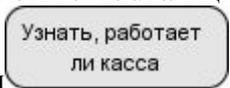


Рис.9 Процесс с точки зрения плательщика коммунальных платежей

Диаграмма, приведенная на рис.9, описывает процесс приема платежей с точки зрения человека, который оплачивает услуги. При помощи основных графических элементов показаны действия плательщика. Нотация BPMN позволила показать и альтернативный ход процесса в случае возникновения ошибочных ситуаций. Рассмотрим более подробно этот процесс.

В процессе участвует два актера – кассир и плательщик. На диаграмме кассир изображен как пул, но подробностей работы кассира диаграмма не раскрывает. Процесс начинается с того, как плательщик, придя в пункт приема платежей, узнает, работает ли этот пункт вообще. Это действие плательщика



показано на диаграмме в виде деятельности . В случае положительного ответа плательщик переходит к следующему действию. Случай, когда касса не работает, показан в виде промежуточного события, прикрепленного к деятельности «Узнать, работает ли касса»



касса не работает. Следующее действие плательщика – это предоставление кассиру квитанции на оплату. Квитанция, предоставленная кассиру, показана на диаграмме



в виде квитанция. Далее, плательщик узнает, возможно ли провести оплату по этой квитанции. Если возможно, то кассиру передается необходимая сумма денег, а он, в свою очередь, отдает плательщику напечатанный чек и сдачу. Данная диаграмма описывает ход процесса приема платежа, однако она не представляет полной картины процесса. На ней не видно ни действий кассира, ни процессов, происходящих с ККМ и ЦБД. Данная диаграмма не является информативной ни для бизнес-аналитика, ни тем более для ИТ-специалиста, который в дальнейшем будет строить архитектуру системы.

При рассмотрении диаграммы, изображенной на рис.8, можно увидеть, что КАССИР является единственным актером процесса, на котором «закрываются» действия всех остальных актеров. Поэтому и диаграмму процесса необходимо строить с точки зрения деятельности кассира.

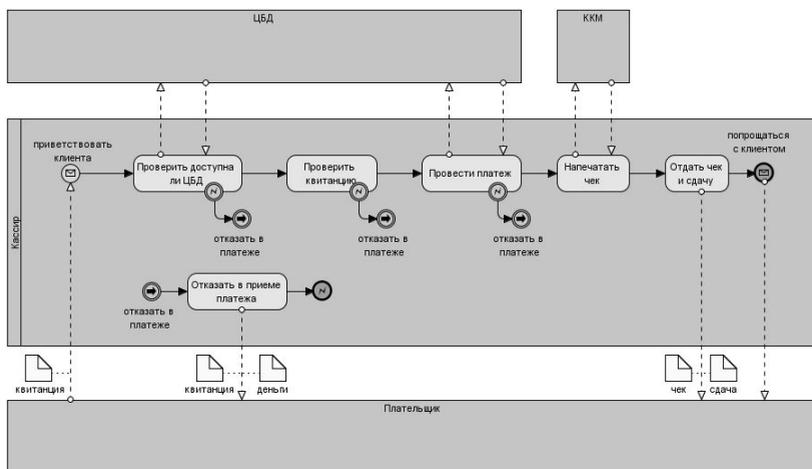


Рис.10 Процесс с точки зрения кассира

На рис.10 видно четыре пула, которые соответствуют «актерам» процесса. Центральное место здесь занимает дорожка с описанием процесса, за который отвечает кассир. Любая ошибка, произошедшая при работе кассира, будь то сбой связи с центральной базой данных, либо перебой со светом, приводит к действию «Отказать в приеме платежа». Для того чтобы не загромождать диаграмму лишними связями, переход на действие «Отказать в приеме платежа» осуществляется при помощи



отказать в
события ссылки платеже

Все актеры, кроме кассира, изображены на диаграмме в виде так называемых «черных ящиков». Это означает, что на диаграмме не раскрываются детали реализации процессов, хотя сама нотация не запрещает этого. Здесь решение о целесообразности отображения деталей – задача проектировщика.

Данная диаграмма окажется полезной и понятной бизнес-аналитику – на ней прозрачно показан весь бизнес-процесс приема платежа, причем диаграмма лишена деталей конкретной реализации, которые зачастую только мешают понять суть процесса. Также диаграмма окажется полезной и информативной для ИТ-специалиста. Данная диаграмма позволит архитекторам и проектировщикам информационной системы правильно спроектировать ее, проанализировать нагрузку, которая может быть оказана на систему и выявить «узкие места». Например, поскольку кассир работает с Центральной Базой Данных и ККМ, то можно сразу сказать, что одним из самых узких мест системы будет обеспечение транзакции проведения платежа в ЦБД и на ККМ. Исходя из такого анализа, можно далее принимать решения о технической реализации процесса.

Итак, в работе представлен подход к проектированию систем массового обслуживания. Для построения информационной модели системы приема коммунальных платежей сначала был

использован подход, основанный на диаграммах прецедентов UML. Полученные в ходе работы диаграмма прецедентов (рис. 8) и сценарий (таблица 1) позволили выделить основные функциональные объекты системы. Затем эти объекты были взяты за основу при построении диаграммы бизнес-процесса приема коммунальных платежей. В итоге была получена модель (рис. 10), которая может быть использована как исходная точка при построении других подобных моделей, то есть был получен паттерн модели приема коммунальных платежей.

Таким образом, можно выделить основные шаги при проектировании систем массового обслуживания.

- Определение основных элементов модели и описание взаимодействий между этими элементами посредством Use Case диаграмм языка UML.
- Определение потока действий, набора событий системы и описание их в нотации BPMN.
- Четкое определение, какой процесс будет центральным при описании всей системы в целом.

Литература

1. Открытые системы [Электронный ресурс] / *Избранные паттерны BPM*; А. БЕЛАЙЧУК
<http://www.osp.ru/os/2009/01/7197856/>
2. ИТ блокнот Николай Войнов [Электронный ресурс] / НИКОЛАЙ ВОЙНОВ. *Практика использования BPMN*
<http://nvoynov.blogspot.com/>
3. ФАУЛЕР М. *UML. Основы, 3-е издание.* – СПб: Символ-Плюс, 2007. – С. 123-129.
4. ALISTAIR COCKBURN, “*Writing Effective Use Cases*”, Addison-Wesley, 2001
5. STEPHEN WHITE, “*Introduction to BPMN*”, BPM Architect, IBM October 16, 2006 © IBM Corporation IBM Software Group
<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>