

УДК 330.46

И.В. Артамонов

ФГБОУ ВО «Байкальский государственный университет», Иркутск,
e-mail: ivan.v.artamonov@gmail.com

ОСОБЕННОСТИ ПРИМЕНЕНИЯ АЛГОРИТМИЧЕСКОГО ПОДХОДА К МОДЕЛИРОВАНИЮ БИЗНЕС-ПРОЦЕССОВ

Ключевые слова: бизнес-процесс, алгоритм бизнес-процесса, алгоритм программы, моделирование бизнес-процессов.

Моделирование бизнес-процессов – неотъемлемый этап внедрения современных технологий и методов на предприятии. Зачастую эта задача требует представления процессов в виде иерархии формализованных операций, описание которых ляжет в основу алгоритмов работы информационных систем или служебных инструкций работников. Зная, что построение схемы для известного алгоритма программы – тривиальная задача, аналитики таким же образом часто пытаются отобразить выполняемые на предприятии бизнес-процессы. Тем не менее, алгоритм выполнения бизнес-процесса существенно сложнее и менее очевиден алгоритма работы программы, и традиционная практика абстрагирования от специфичных бизнес-процессу деталей в пользу формальных алгоритмических конструкций часто приводит к снижению адекватности полученной модели. Статья предлагает изучить факторы, от которых зависит возможность формального представления процессов без потери необходимых для моделирования деталей. Проводится сравнительный анализ применения алгоритмизации к разработке программ и проектированию бизнес-процессов. Последовательно исследуются возможности проверки алгоритмов, описания их участников и операций, возможности обратимости результатов работы алгоритма, оценки его сложности, времени выполнения и стоимости. Проведенный анализ позволяет предложить классификацию видов процессных алгоритмов и описать особенности их моделирования. В зависимости от вида процесса можно сделать вывод о применимости к его модели различных способов анализа и информатизации.

Введение

В основе функционирования любого современного предприятия лежит, безусловно, процессный подход. Он предполагает смещение акцентов от управления отдельными структурными подразделениями к управлению сквозными бизнес-процессами, охватывающими все предприятие и даже выходящими за его пределы. Поэтому в основе подхода лежит деятельность по выявлению, описанию и исполнению бизнес-процессов компании. Под бизнес-процессом будем понимать совокупность взаимосвязанных процедур и действий, совместно достигающих определенной бизнес-цели [1]. Обычно под целью понимается результат в виде определенного продукта, полезного для клиента бизнес-процесса.

В условиях жесткой конкуренции, динамичности рынков и внешней среды

хорошим преимуществом является использование информационных технологий для автоматизации различных сфер деятельности [2-4]. Все современные информационные системы (ИС) предприятия тоже ориентированы на процессную модель. Таким образом, деятельность по выявлению, формализации и структуризации бизнес-процессов является критической для внедрения практически любых информационных систем, а, значит, важной для функционирования и успешной конкурентной борьбы предприятия [5].

Построение процессной модели сопровождается разработкой и анализом схем (моделей) целевых бизнес-процессов. Часто этот этап рассматривают как промежуточный перед внедрением ИС, а выполняется он графическим образом. В зависимости от множества факторов основными инструментами разработки

моделей выступают как обычные офисные пакеты программ с возможностями только отрисовки несложных схем, так и специализированные комплексы для моделирования процессов, возможно, даже ориентированные на стандарты BPMN, UML, EPC, ARIS, IDEF и пр. Такие комплексы могут быть встроены во внедряемые информационные системы, что облегчает процесс применения полученных схем на практике, либо поставаться и использоваться отдельно, что удешевляет этап, но приводит к необходимости конвертации моделей процессов к реальности.

Определение бизнес-процесса дает увидеть важный вывод: незавершенная операция по созданию продукта практически всегда не имеет смысла. Человеческий труд, работа множества контрагентов, затраты ресурсов и энергии оказываются бесполезными, если конечный результат не был получен, бизнес-процесс не достиг цели.

Компания должна ориентироваться на многократное выполнение бизнес-процесса, возможно, с разными входными ресурсами, но которой всегда создаст предполагаемый результат. Поэтому вторая опасность незавершенности состоит в том, что неожиданная остановка процесса может нарушить возможности многократного выполнения и, тем самым, остановит бизнес.

Устранение проблемы остановки состоит в возврате системы, всех ее участников и ресурсов в некоторое начальное состояние, из которого она может продолжить работу и начать процесс заново. Оснащение подобными возможностями «перезагрузки» процессов взаимодействия независимых участников нередко представляется более сложной задачей, на первой взгляд, неактуальной и невозможной. Так как, следуя существующим техникам моделирования, бизнес-процесс выполняется по строгому алгоритму, на вход которого подаются ресурсы, а результатом является продукт.

Поэтому проектировщики процессов и системные аналитики упускают важные свойства, в корне отличающие алгоритм выполнения бизнес-процесса от алгоритма в программной инженерии. Если учитывать эти свойства, область моделирования бизнес-процессов оказывается

более сложной и более ответственной, нежели чем принято считать.

Поэтом целью статьи является сравнительный анализ принципов применения алгоритмического подхода в процессном управлении и программной инженерии, в результате исследования разработан перечень найденных различий, показана актуальность их знания для практической деятельности и предложена классификация процессов по особенностям методики их моделирования.

Анализ применения алгоритмизации к бизнес-процессам и программам

Проверка алгоритма. Алгоритм кодируется на определенном языке программирования и выполняется аппаратурой. Авторами алгоритма выступают программисты, специализирующиеся на формальном представлении задачи. Перевод алгоритма на машинный язык сопровождается различными операциями проверки, и полученный двоичный код готов к многократному выполнению. Правильность записи алгоритма контролируется семантикой языка программирования и развитыми средствами тестирования и верификации. А современная аппаратура, выполняющая программу, настолько надежна, что вероятность неправильного исполнения правильного записанного алгоритма ничтожно мала.

Алгоритм бизнес-процесса практически всегда записывается в графическом виде. Авторами модель процесса часто являются аналитики, специалисты предметной области, которые в дальнейшем не будут заниматься его кодированием в программе. Полученная модель процесса не содержит переменных, функций, точных выражений и других инструментов, сшивающих его участки воедино. Средства графического моделирования практически не контролируют никакие аспекты, кроме, возможно, самых примитивных, например, наличия входов / выходов. Ошибиться в графическом описании намного проще, чем при записи алгоритма на строгом языке. Единственным решением проблемы остается то, что графическая модель часто впоследствии интерпретируется в программный код и возможные ошибки исправляются. Однако верификация на детальном уровне не устраняет

проблем глобального, и ошибка алгоритма возникнет на стыке нескольких правильно работающих программ.

Существуют средства моделирования, встроенные в программные продукты выполнения бизнес-процессов. В таких средствах осуществляется проверка и верификация графических моделей. Однако они ориентированы на ограниченную область контроля, и надежно описывать процессы взаимодействия нескольких независимых предприятий (реально независимых, а не являющиеся филиалами или вспомогательными организациями) с их помощью сложно. Тем более, при внедрении подобных систем первичная настройка бизнес-процессов осуществляется по моделям, созданных каким-либо другим, зачастую неформальным способом. Использование формальных средств моделирования, например, сетей Петри, или универсальных систем имитационного моделирования [6] повышает качество проектирования, однако требует соответствующей методической подготовки аналитика. А существующие средства имитационного моделирования бизнес-процессов не учитывают множество важных нюансов, как показано в [7].

Исполнители алгоритма. Алгоритм программы можно разделить на несколько частей. Каждая часть обычно выполняется отдельной подпрограммой, намного реже – программой на отдельном компьютере. В любом случае, поведение (под)-программы доподлинно известно и предсказуемо. Перечень возможных проблем известен программисту, и он всегда уверен в том, что (под)-программа будет возвращать понятный результат. Центральная программа руководит всеми своими сателлитами, может влиять на их работу и определяет операции, ими выполняемые. А в случае с распределенной системой поведение ее подсистем жестко контролируется опубликованными или заранее обговоренными интерфейсами.

В бизнес-процессе участвует некоторое множество партнеров. Они бывают настолько независимы, что границы их открытости лежат в пределах коммерческой тайны, а в других плоскостях они могут быть даже конкурентами. Не может быть и речи о допуске одно-

го партнера к ресурсам другого. Работа осуществляется строго в пределах оговоренных контрактов, выполнение которых контролируется каждым в отдельности. То, как реализовывать контракт, тоже является прерогативой партнера, особенности его работы скрыты от остальных.

Предсказуемость результата. Алгоритм программы может предполагать ввод информации пользователем. Он может менять поведение программы в зависимости от введенных данных пользователем. Но никогда алгоритм программы не содержит действий пользователя, которые никоим образом не контролируются со стороны вычислительной техники. Поэтому результат работы программы всегда предсказуем, несмотря на любое поведение пользователя.

Алгоритм бизнес-процесса, напротив, может содержать указание для выполнения набора ручных операций или сам перечислять эти операции. При проектировании иногда переход между операциями, выполняемыми вручную, и автоматизированными операциями вообще нельзя четко проследить на рассматриваемом уровне абстракции процесса. Ручные операции могут иметь непредсказуемый результат, зависят от квалификации оператора и, вообще, человеческого фактора. Неправильная интерпретация служебных инструкций, забывчивость или рассеянность могут спровоцировать ошибки, которые останутся невыявленными и трудно отслеживаемыми ввиду своей физической или ручной природы.

Обратимость результатов работы. Алгоритм программы зачастую работает с объектами информационной природы. Он получает данные на вход и создает новые данные. Результаты работы такой программы легко отменить – можно удалить новые данные, уничтожить созданный объект, очистить экран. В крайнем случае, можно вернуть механизм в начальное положение.

Алгоритм бизнес-процесса воспринимает и информацию, и материальные объекты одинаково – как ресурсы или результат работы. В отличие от информации, ресурсы имеют свойство истощаться и необратимо односторонне преобразовываться. Поэтому цена ошибки

бизнес-процесса намного выше, чем цена ошибки рядового алгоритма. Потраченные и поглощенные ресурсы невозможно вернуть в изначальное состояние, созданную смесь не разделить по составляющим или не забрать рассеянную в тепло энергию. Мало того, промежуточные результаты неуспешного процесса невозможно просто «удалить». Скорее всего, необходимо ввести специальные операции, которые устраняют последствия неудачи, «компенсируют» результаты работы, тратя очередные ресурсы. И, возможно, система будет возвращена не в начальное состояние, а какое-то наиболее «похожее» [8]. Бывает также, что разработка таких «обратных» операций сложнее и более трудоемка, чем «прямых», поэтому эту проблему предпочитают «не замечать» и отдавать решение в руки администраторов системы [9].

Время выполнения. Алгоритмы программ обычно приспособлены к быстрому выполнению или череде выполнений между реакциями пользователей. В ожидании события со стороны пользователя программа может находиться бесконечно долго, по большей части, не влияя на производительность остальной системы. Любое событие провоцирует каскад реакций, в результате которого программа либо завершает работу, либо ожидает следующего события. Все операции, производимые компьютером, программируются как быстрые и точные. Поэтому можно блокировать ресурсы, которые в данный момент используются, получать более высокий приоритет работы на короткое время, передавать данные другой, параллельно ожидающей программе, и использовать иные механизмы, хорошо работающие в небольшой отрезок времени.

Алгоритм бизнес-процесса может выполняться несколько недель. Бесспорно, его автоматизированные части также способны к практически мгновенному исполнению. Однако остальная часть состоит из ручных операций, длительного ожидания завершения других процессов, использования ресурсов разной природы и качества. Используемые ресурсы нельзя заблокировать, параллельно запущенная программа не сможет находиться в состоянии ожидания данных

бесконечно долго (ее придется перезапустить), а другие программы многократно перезапускаются, что, например, делает неудобным использование примитивов синхронизации операционной системы. В целом проблематика надежного выполнения длительных бизнес-процессов является актуальной, важной и пока не решенной до конца, несмотря на множество попыток перенести идеи транзакционного управления в эту область ([8], [10–15]).

Стоимость выполнения. Обычно запуск алгоритма программы не требует большого количества ресурсов. Немного повышается загрузка процессора, систем памяти. Возможно, будет привлечен пользователь или запущен какой-то механизм. Но в самом общем варианте однократный запуск типичной программы несколько увеличит расход электроэнергии.

Запуск бизнес-процесса практически всегда вовлекает нескольких исполнителей и множество ресурсов. Даже если ресурсами и результатом будет информация, то зачастую она создается несколькими пользователями, время работы которых может стоить дорого, не говоря уже о материальных истощаемых ресурсах и рисках процесса быть незавершенным. Все это приводит к тому, что однократное выполнение процесса намного дороже запуска программы. А в купе с длительным выполнением все это существенно ограничивает возможности тестирования (а особенно тестирования «обратного», восстановительного процесса) алгоритма, что негативно влияет на его работоспособность.

Стандарт записи. Программный код в стандартном, императивном подходе состоит из множества типовых команд и конструкций. Практически во всех популярных языках программирования есть итерации, условия, переменные, подпрограммы и т.д. Существуют, видимо, даже негласные правила отображения блоков выбора, циклов, ввода/вывода данных. Язык алгоритмизации прост, универсален и преподается на первых курсах профессионального образования. Этот язык позволяет программистам легко понимать программы друг друга и алгоритмы, записанные любым способом.

Основными нотациями, применяемыми в России, являются UML, EPC, BPMN или IDEF. В рабочей среде зачастую практикуют собственноручно созданные стандарты и отрисовку с помощью офисного программного обеспечения. Не все общепризнанные стандарты способны верифицировать модель бизнес-процесса, а те, которые способны это делать, зачастую используются (и преподаются) таким образом и в таких средах, что проверку модели осуществить невозможно. Последнее десятилетие, с приходом стандартов BPMN, BPEL и подхода BPM (управления бизнес-процессами) [16] ситуация исправляется: графическую схему процесса в нотации BPMN легко можно отобразить на псевдо-программный код на языке BPEL и даже запустить выполнение [17]. Однако это требует высокой квалификации аналитика и специализированного ПО.

Таким образом, на сегодняшний день существует несколько подходов к моделированию бизнес-процессов, и использование их предопределяется множеством факторов. Вкупе с другими существенными ограничениями и сложностями в проектировании алгоритмов процессов это приводит к закономерному результату: в большинстве случаев, независимо от размеров и возможностей компании процессы в ней описываются наиболее простым и дешевым способом на неформальном языке и в программе, которой обладают все заинтересованные стороны. Естественно, программа моделирования не обладает возможностями для верификации моделей бизнес-процесса и не учитывает множество его нюансов.

Результаты исследования и их обсуждения

Результаты приведенного исследования суммированы в таблице.

Таблица

Свойство	Компьютерная алгоритм	Модель бизнес-процесса	Свойство
Проверка записи	Многочисленная, средствами языка программирования, трансляторами и средой выполнения	Ограниченная, только при использовании специализированного ПО	Проверка записи
Участники	Зачастую зависимые, подконтрольные, со строго регламентированным поведением	Зачастую независимые, неподконтрольные	Участники
Операции	Заранее определенные и запрограммированные, с предсказуемым результатом	Автоматизированные или ручные, могут быть с непредсказуемым результатом	Операции
Обратимость эффектов	Зачастую возможна и легко реализуема	Зачастую невозможна, требуется введение специальных компенсирующих действий для минимизации последствий	Обратимость эффектов
Время выполнения	Быстрое, возможны блокировки ресурсов, использование стандартных техник системного и сетевого программирования	Длительное, зачастую блокировки ресурсов невозможны, ПО должно быть рассчитано на многократный перезапуск и выполнение одного и того же процесса	Время выполнения
Стоимость выполнения	Небольшая	Высокая в связи с затратой невозможных ресурсов	Стоимость выполнения
Описание алгоритма	Формальное, однозначное, понятное большинству заинтересованных лиц	Зачастую неформальное, неоднозначное, но понятное большинству заинтересованных лиц	Описание алгоритма
Стандарт записи алгоритма	Нет, но и не требуется, так как алгоритм вручную отображается в программный код.	Есть, но слишком большое разнообразие.	Стандарт записи алгоритма

Существуют ситуации, когда алгоритмы бизнес-процессов в точности похожи на алгоритмы программ и наоборот. Но заметим, во-первых, ситуации, когда алгоритм программы выполняют независимые участники с неотменяемыми результатами, являются скорее исключением из общей практики. Во-вторых, такие программы являются в первую очередь координаторами бизнес-процессов, а, значит, выполняют его алгоритм.

Обратная ситуация, когда бизнес-процесс похож на алгоритм программы, безусловно, возможна.

Поэтому предложенные отличия предлагают создать группировку бизнес-процессов на три класса: схожие по всем признакам с алгоритмами программ, не схожие с ними, но подлежащие надежному проектированию, и третий класс, не удовлетворяющий этим свойствам.

Первый тип можно проектировать и разрабатывать как обычные программы. Этот класс называется бизнес-операциями, является элементарным, самым детальным уровнем декомпозиции, и из него строятся все остальные классы процессов.

Второй класс включает процессы первого, может длительно выполняться, требует ручных операций, иных механизмов контроля ресурсов и выполнения. Подтверждать его работоспособность можно только в специальных средствах моделирования, встроенных в программное обеспечение для выполнения бизнес-процессов. И существует программное обеспечение, способное транслировать процессы второго типа в программный код.

Третий класс похож на второй, однако не может быть надежно спроектирован, так как затрагивает участников и процессы, выходящие за пределы контроля моделирующего ПО. Например, часть подпроцессов выполняется вручную или сторонним программным обеспечением.

Второй и третий типы представимы как один в случаях, если:

– В компании не используются программные системы, способные к моделированию и выполнению бизнес-процессов.

– Моделирующая система не способна верифицировать модель, оценивать ее качество и надежность, т.е. она выполняет только роль графического редактора.

– Проектируемые бизнес-процессы лежат вне границ контроля моделирующей системы (например, CRM-система, которая способна выполнять процессы по схеме, не отвечает за вопросы формирования бухгалтерской отчетности).

Как видно, вероятность того, что компания не методично подходит к проектированию сложных бизнес-процессов, достаточно высока. Тем более, развитие партнерских отношений, технологий интеграции и специализация бизнеса ведут к тому, что процессы все чаще выходят за границы предприятия.

Проблемы, связанные с проектированием и моделированием бизнес-процессов, проистекают часто из неверного понимания специфики алгоритмов процессов второго и третьего типов. Они сложнее, неудобнее и непривычнее к точной разработке, которая, вдобавок, слабо автоматизирована, чем типичное программирование

Заключение

Описание алгоритма выполнения бизнес-процесса методологически более сложная деятельность, чем работа над алгоритмом программы. Большинство бизнес-процессов, которые моделируются для целей внедрения информационных технологий, относятся ко второму или третьему классу. Первый класс, бизнес-операции – это своеобразный мост между алгоритмом функционирования бизнеса и алгоритмом работы программы. Программа становится технологическим воплощением простейших, неручных бизнес-операций.

Моделирование процессов по устойчивейшей практике начинается «сверху-вниз», и между простейшими операциями и процессами верхнего слоя абстракции лежат уровни декомпозиции, которые должны быть рассмотрены и проанализированы аналитиком. Он должен принимать во внимание все описанные модельные свойства бизнес-процесса и доводить декомпозицию для приемлемого уровня, понятного неспециалисту предметной области.

На практике аналитик может не обладать достаточной квалификацией и опытом. Такие проблемы, как обратимость (компенсируемость) эффектов бизнес-процессов, длительность выполнения работ, неподконтрольность исполнителей и пр. могут игнорироваться и перекладываться на плечи технических исполнителей. Хотя это недопустимо. Например, программист в ходе своей работы столкнется с проблемой обработки исключительных ситуаций и отмены результатов незавершенного процесса, другими словами с проблемой компенсации процесса. Не владея ни логикой всей операции, ни достаточным знанием предметной области, он не сможет правильно реализовать сложную процедуру «отката», которая, к тому же, может за-

трагивать зоны ответственности других разработчиков.

Бизнес-процесс, особенно связывающий независимых партнеров, должен определять развитие и разработку программ, глобально решать свои цели, обеспечивать всеобъемлющее восстановление после сбоя и «откат» к начальному состоянию. Здесь в полной мере проявляется свойство эмерджентности систем: сумма составляющих не равна всей системе. Компоненты, работая по отдельности, преследуя локальные цели, не способны решить задач целостной системы. Так и в бизнес-процессе: необходима точка зрения, направляющая каждый элемент в отдельности. Плохое проектирование бизнес-процесса лишает глобальную точку зрения адекватности.

Библиографический список

1. Workflow Management Coalition. Workflow Management Coalition Terminology & Glossary. Issue 3.0. Winchester, 1999. – 65 p.
2. Пешкова О.В., Миненков Р.В. Анализ модели ИТ-эффекта на основе структурного подхода // Прикладная информатика. – 2014. – № 5 (53). – С. 5–15.
3. Балашова Н.В., Репина П.В. Опыт внедрения сбалансированной системы показателей // Известия ИГЭА (БГУЭП). – 2011. – №6. – С. 45.
4. Антипина П.В. «Облачные» технологии как инструмент логистического аутсорсинга // Трансформация логистических систем в современных условиях. – 2015. – С. 5-9.
5. Хитрова Т.И., Багдueva Т.И. Проблемы оценки степени готовности системы управления к внедрению информационных технологий // Активизация интеллектуального и ресурсного потенциала регионов: новые вызовы для менеджмента компаний: материалы конференции. – Иркутск, 2016. – С. 167–172.
6. Матусевич Д.С. Применение ресамплинга для имитационного моделирования систем массового обслуживания // Известия ИГЭА (БГУЭП). – Иркутск, 2013. – № 1. – С. 28.
7. Wil M.P. van der Aalst. Business Process Simulation Survival Guide // Handbook on Business Process Management / Jan vom Brocke, Michael Rosemann. Berlin : Springer-Verlag. – 2015. – P. 337–370.
8. Артамонов И.В. Моделирование бизнес-транзакций. – Иркутск: Изд-во БГУ, 2016. – 190 с.
9. Greenfield P., Fekete A., Julian J. Compensation is Not Enough // Proceedings of the 7th International Conference on Enterprise Distributed Object Computing (EDOC) 2003. – Brisbane, Australia, 2003. – 232 p.
10. Alonso G., Divyakant A., et. al. Advanced Transaction Models in Workflow Contexts // ICDE '96 Proceedings of the Twelfth International Conference on Data Engineering. – 1996. – P. 574–581.
11. Doreen T. An analysis of transactions in service-centric systems [Электронный ресурс] // Department of Mathematics and Computing Science Rijksuniversiteit Groningen. – URL: <http://www.cs.rug.nl/~aiellom/tesi/tuheirwe.pdf>
12. Grefen P.A. Taxonomy for Transactional Workflows. CTIT Technical Report 02-11. – Enschede: University of Twente, 2002. – 20 p.
13. Grefen P. Transactional Workflows or Workflow Transactions? // DEXA '02 Proceedings of the 13th International Conference on Database and Expert Systems Applications, 2002. – P. 60–69.
14. Jajodia S., Kerschberg L. Advanced Transaction Models and Architectures. – New York: Springer, 1997. – 379 p.
15. Sheth A., Devashish W. Transactions in transactional workflows // Advanced Transaction Models and Architectures / Larry Kerschberg, Sushil Jajodia. – New York : Springer, 1997. – 379 p.
16. Smith H., Fingar P. Business Process Management: The Third Wave. Meghan-Kiffer Press, 2003. – 311 p.
17. Артамонов И.В. Описание бизнес-процессов: вопросы стандартизации // Прикладная информатика. – 2011. – № 3. – С. 20–28.