

ПРИНЦИПЫ ФОРМАЛЬНОГО ПРЕДСТАВЛЕНИЯ ПОВЕДЕНЧЕСКОЙ ПЕРСПЕКТИВЫ МОДЕЛИ БИЗНЕС-ПРОЦЕССА

И.Г. Федоров,

кандидат технических наук, профессор Московского государственного университета экономики, статистики и информатики (МЭСИ)

E-mail: IFedorov@mesi.ru

Адрес: г. Москва, ул. Нежинская, д. 7

В работе анализируются ограничения отдельных моделей и нотаций описания бизнес-процессов. Показано, что ни один из рассмотренных видов диаграмм не позволяет одновременно отобразить изменение во времени всех сущностей, необходимых для описания динамики поведения системы. Установлено, что поток управления образуется в результате движения объекта управления, последний определяет статус исполнения на некотором интервале времени. Для описания сложных процессов следует применять вложенные сети Петри, причем каждая подсеть связана со своим объектом управления. Во избежание коллизий недетерминированного поведения следует использовать сохраняющие сети Петри. Предлагаемый в этой работе подход к проектированию исполняемой модели процесса предполагает последовательное моделирование информационной и поведенческой перспектив процесса, взаимное согласование частных моделей.

Ключевые слова: диаграммы состояния, потоков работ и данных, управления, сети Петри, нотации EPC и BPMN.

1. Введение

Проблема построения точного и полного описания работы бизнес-процесса актуальна так же, как и построение алгоритма в программировании. Часто аналитическое моделирование процессов заканчивается лишь некоторой картинкой, которая не вполне точно передает поведение исследуемой системы. Однако исполняемая модель бизнес-процесса должна реализовывать

полный и точный алгоритм работы системы, в математическом смысле этого слова [1].

Предметом дальнейшего анализа будет поведенческая перспектива модели процесса. Она описывает динамику системы, определяет порядок выполнения работ [2]. В теории систем для описания поведения принято использовать диаграмму состояний — STD [3], но в бизнес-информатике для моделирования процессов используются диа-

граммы разных типов: потоков данных — DFD [4], потоков работ — WFD [5], управления — CFD [6], блок-диаграммы — Flowchart [7], событийно-ориентированные цепочки процессов — EPC [8], сети Петри [9], диаграммы BPMN [10]. Возникает вопрос, насколько эти формализмы моделирования подходят для описания поведенческой перспективы и адекватны поставленной задаче построения полного и точного алгоритма работы бизнес-процесса?

Цель настоящей работы заключается в исследовании ограничений отдельных моделей и нотаций, используемых для описания бизнес-процессов, формулировании принципов, которые помогут создать исполняемые модели, описывающие алгоритм работы процесса.

2. Диаграмма состояний

Диаграмма состояний (STD) есть традиционный способ описания поведения системы. Принято выделять управляющую и управляемую части системы, управляющее и вычислительное состояния [3]. Например, управляющее состояние «Процесс работает / закончен» отражает статус всей системы как целого. Вычислительное состояние связано с объектом управления, изменение статуса которого отражает ход процесса. Например, изменение характеристик информационного объекта «кредитная заявка» определяет ход процесса выдачи банковского кредита. Предметом дальнейшего рассмотрения станут вычислительные состояния.

Поскольку внутри большой прикладной программы могут существовать сотни и даже тысячи переменных и несколько потоков управления [11], принято выделять переменные состояния [12]. Для упрощения анализа, в каждый момент времени рассматривают изменения одной переменной состояния, которая определяет состояние всей системы на данном этапе ее функционирования [3].

Чтобы понизить размерность решаемой задачи предлагается рассматривать диаграмму состояния сложной системы как иерархически вложенную и составную [13]. В первом случае принято говорить о декомпозиции переменной состояния, причем выделяют количественные и качественные состояния объекта управления. Качественное состояние называют комплексным, если его поведение можно декомпозировать на отдельном листе диаграммы, где будут рассматриваться только изменения количественных параметров переменной состояния. Во втором случае

весь процесс разделяется на этапы, в каждом из них свой объект управления, т.ч. составная диаграмма объединяет в цепочку несколько простых. Говоря о составной диаграмме, мы будем обращать внимание на смену переменной состояний, которая определяет границы этапов исполнения процесса.

Выделим принципы проектирования диаграммы состояний: отбирать ограниченное число переменных состояния, на каждом временном этапе исполнения процесса выделять одну из них, контролировать смену переменных, это позволяет выделить этапность процесса. Создавать составные диаграммы для независимых переменных состояния. Следует разделять качественные и количественные изменения переменной состояния, это позволяет построить иерархическую декомпозицию диаграммы состояний.

3. Диаграмма потоков данных

Диаграмма потоков данных (DFD) используется для описания движения документов и обработки информации [4]. Ее принято называть трансформирующей, т.к. она изображает работы, которые преобразуют входной поток, но не показывают те, которые его не изменяют [14]. При этом изображаются только данные, непосредственно участвующие в преобразовании, а те, которые не используются в операции, на схеме не показываются. На вход узла поступает ровно столько данных, чтобы сгенерировать выходной поток. Последний без изменений передается на вход следующего узла. Поток может обогащаться данными из внешних источников или обедняться, если информация записывается на внешние хранилища и в последующих работах не используется. Справедлив закон сохранения данных: сумма всех входящих потоков узла равна сумме исходящих. Этот же принцип применим для всего процесса в целом.

Состояние системы определяется статусом объекта управления, а не выполняемой работой. Однако диаграмма данных не показывает изменение объекта управления. Она изображает работы, приводящие к смене объекта управления.

Сформулируем принципы моделирования диаграммы потоков данных. Трансформирующая диаграмма показывает только работы, которые изменяют информационный поток, и лишь те данные, которые непосредственно участвуют в обработке. Информационный поток является сохраняющим,

он не генерирует новые данные и не уничтожает существующие. Новый объект управления образуется в результате преобразования старого, при этом в него переносится вся необходимая информация.

4. Диаграмма потоков работ

Диаграмма потоков работ (WFD, Flowchart) изображает порядок выполнения операций процесса [15]. Узлами диаграммы являются работы, а дуги определяют временную очередность их выполнения, они обозначают не поток объектов, а поток управления. К сожалению, сущность этого понятия не определена. В программной инженерии под потоком управления понимается множество всех возможных путей исполнения программы. В бизнес-информатике диаграмма работ часто показывает отдельные, наиболее вероятные варианты исполнения, забывая про редкие сценарии и исключительные ситуации (в этом случае перед аналитиком не ставится задача полноты разрабатываемой модели).

Диаграмму работ, по аналогии с диаграммой потоков данных, называют трансформирующей [23], что не верно, поскольку она изображает не только операции, преобразующие входной поток, но и работы, которые данные не изменяют, но маршрутизируют. Например, логические операторы, которые графически изображают функцию принятия решения, поток не модифицируют, но направляют в нужном направлении. Сравним диаграммы потоков данных и работ: на обеих узлы изображают работы процесса, но на первой дуге соответствует информационный поток, а на второй — «материальное» содержимое дуги не определено. Иногда подпись к дуге диаграммы работ обозначает состояние некоторого объекта после обработки; можно предположить, что это переменная состояния.

Выделим совокупность операций, которые связаны общим объектом данных. Будем называть его объектом управления, если он сохраняет результат текущей операции и передает следующей. Объект управления последовательно трансформируется по мере исполнения операций процесса. На некотором интервале времени его состояние определяет статус исполнения процесса. Если возникает необходимость смены объекта, новый объект не возникает ниоткуда, он получается из старого путем копирования всех важных данных. Рассмотрим пример процесса «от заказа до продукта». На первом этапе

процесса объектом управления является «Заявка». На следующем этапе происходит оплата, основным становится информационный объект «Счет». Новый объект возникает в результате трансформации исходного, путем переноса данных из «Заявки» в «Счет». Когда происходит отгрузка товара, основным объектом становится «Накладная». Смена объекта управления означает, что единый процесс делится на сеть подпроцессов. Следует утверждать, что поток управления образован движением объекта управления.

Тем не менее, встречаются аналитические модели, где две смежные операции не связаны между собой ни общим потоком информации, ни объектом, над которым выполняется работа. Например, первая работа обрабатывает «Заказ», следующая — «Продукт» и далее — «Счет». Человек может домыслить отсутствующие на схеме правила преобразования этих объектов исходя из контекста задачи. Однако, бесконтрольная смена объекта управления абсолютно неприемлема в исполняемой модели бизнес-процесса, поскольку машина не сможет восстановить отсутствующие на схеме детали. В ситуации хаотической смены объекта управления, становится трудно определить текущее состояние системы.

Диаграммы работ пришли из программной инженерии, где они используются для описания алгоритма программы. При этом система подвергается последовательному анализу с использованием моделей состояний, потоков данных, сущность-связь. И только согласованный результат моделирования превращается в диаграмму потоков работ. В случае же описания бизнес-процессов предварительное моделирование данных и состояний не производится. Бизнес-аналитики моделируют диаграмму работ в обход системного анализа, что приводит к неожиданным результатам.

Выделим основные принципы моделирования диаграммы работ:

- ♦ следует выделить совокупность операций, которые связаны общим объектом данных, который сохраняет результат исполнения текущей операции и передает следующей;
- ♦ диаграмма работ должна быть «согласована» с диаграммой состояний;
- ♦ поток управления образуется в результате движения объекта управления;
- ♦ смена объекта управления может происходить только под контролем аналитика.



Рис. 1. Диаграмма EPC

5. Диаграмма EPC

В качестве примера, рассмотрим диаграмму EPC [8], которая относится к классу потоков работ, описывает порядок операций процесса и оперируют основными понятиями «Работа», «Событие» и «Ветвление». Методология ARIS различает начальное, конечное и промежуточное события процесса. Начальное событие трактуется как внешнее, инициирующее данный процесс, а конечное используется для запуска следующего процесса в цепочке. Напротив, промежуточное событие, вопреки названию, используется для фиксации состояния объекта в результате исполнения очередной операции [16]. Таким образом, диаграмма EPC неожиданно обнаруживает характерные черты диаграммы состояний (STD), показывает очередность работ и вызванную ими смену состояний (рис. 1).

В этом заключается сила и преимущество диаграммы EPC: она фиксирует внимание аналитика на изменении состояния объекта управления. К сожалению, методология ARIS [10] не обращает внимание на смену объекта управления, при этом на одной диаграмме могут существовать объекты, которые не могут быть получены путем трансформации одного в другой (например, информационный объект «Заказ» не может быть трансформирован в материальный «Продукт»). В результате аналитик изображает на схеме один процесс, тогда как на самом деле он имеет дело с цепочкой взаимодействующих подпроцессов.

6. Диаграммы управления

В программной инженерии диаграммой управления (control flow) принято называть множество всех возможных путей исполнения программы, представленное в виде графа. Каждый узел диаграммы соответствует базовому блоку — прямолинейному участку кода, не содержащему в себе ни операций передачи управления вне этого блока, ни точек, на которое управление передается извне. Вход в базовый блок возможен только через первую инструкцию, а выход через последнюю. Направленные дуги, связывающие узлы, указывают переход управления [17].

При моделировании бизнес-процессов часто смешивают диаграмму управления и потоков работ, однако они не эквивалентны. Диаграмма управления шире, чем граф очередности работ, уточняет его в части расписания исполнения и бизнес-правил, имеет детализацию уровня действий, описывает все варианты исполнения [18]. Иначе говоря, диаграмма управления описывает технологию, способ, гарантирующий достижение запланированного результата, при условии точного выполнения заранее определенного набора действий. Вместе с тем, диаграмма управления унаследовала недостаток графа потоков работ, она не отражает текущее состояние объектов, его придется подразумевать на подписи к стрелке переходов.

Основные принципы, используемые при моделировании диаграммы управления, заключается в выделении базовых блоков, которые рассматриваются как повторно используемые подпрограммы; обеспечении полноты описания всех возможных маршрутов исполнения; детализации описания работ до уровня элементарных действий.

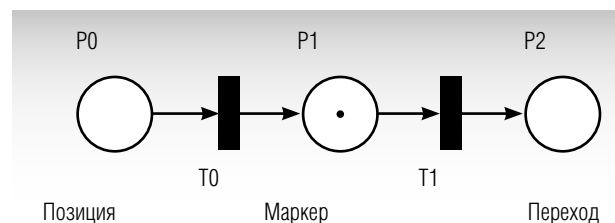


Рис. 2. Сеть Петри

7. Сети Петри

Сети Петри предназначены для моделирования динамического поведения системы. Они представляют помеченный двудольный ориентированный граф, состоящий из вершин двух типов — позиций и переходов, соединенных между собой дугами, причем вершины одного типа не могут быть соединены непосредственно (рис. 2). В позициях могут размещаться маркеры, способные перемещаться по дугам через переходы вдоль сети Петри [9]. Маркер обычно ассоциируется с некоторым объектом — предметом, который имеет материальную природу или информационной сущностью. Переход ассоциируется с событием, работой или перемещением.

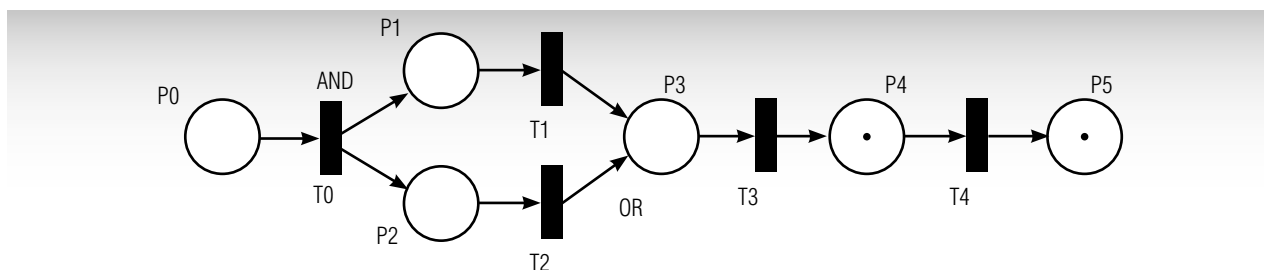


Рис. 3. Процессный паттерн CP10

ем, он изменяет состояние системы путем передвижения маркера из позиции в позицию. Позиция пассивна, она не изменяет и не перемещает маркер, только сохраняет его между двумя переходами.

Состояние сети Петри в любой момент времени определяется расположением маркеров в позициях, а изменение состояния самого маркера в результате прохождения через переход или смена объекта, с которым связан маркер, не анализируется. Но в реальности состояние имеет решающее значение. Рассмотрим примеры. Если заявка возвращена на несколько шагов назад для повторной обработки, то когда она вновь начнет движение вперед, ее маршрут может отличаться от траектории заявки, движущейся по маршруту впервые. К сожалению, сеть Петри затруднительно смоделировать эту ситуацию.

Для анализа бизнес-процессов выделен специальный подкласс – сети Петри потоков работ (WFPN), обладающий следующими свойствами: модель имеет по одной начальной и конечной позиции, называемые истоком и стоком; существует конечный маршрут, ведущий из истока в сток [19]. Такая сеть соответствует определению базового блока (см. выше диаграммы управления). Для процесса, моделируемого WFPN, определены требования бездефектного завершения: (а) в сети нет мертвых (т.е. никогда не срабатывающих) переходов; (б) сеть является безопасной, в позиции может размещаться не более одного маркера; (в) когда маркер достигает стока, в остальных позициях не остается других маркеров.

Моделирование процесса сетями WFPN показывает, что некоторые комбинации логических операторов, которые по отдельности имеют простое поведение, при объединении в последовательности могут создавать коллизии. Например, операторы «И» и «ИЛИ» по отдельности имеют предсказуемое поведение, а когда объединяются вместе, имеют неопределенную семантику исполнения [21]. Рассмотрим паттерн номер CP 10 (рис. 3). Поток

управления вначале разветвляется на узле «AND», при этом, один маркер создает потоки в двух параллельных ветвях. Затем ветви объединяются на узле «OR», который пропускает далее маркеры из обеих параллельных ветвей. Возникает коллизия, один входной маркер порождает несколько выходных, число маркеров увеличивается.

Свойство безопасности не исключает коллизию. Дело в том, что после объединения ветвей, маркеры не останавливаются в первой же позиции, ожидая прихода следующего, а будут продвигаться вперед, так что в любой момент времени маркеры, поступившие из параллельных ветвей, окажутся в разных позициях. При этом безопасность не нарушается, а коллизия остается. Реальным выходом было бы использовать более строгое ограничение – сохраняемость [20], предполагающее, что число циркулирующих в сети маркеров не меняется, они не уничтожаются и не создаются. В рассмотренном примере сеть Петри породила дополнительные маркеры, т.е. не является сохраняющей, хотя остается безопасной.

Авторы WFPN не обращают внимания на смену объекта управления. Мы же предложим рассматривать сеть WFPN как составную, образованную конкатенацией фрагментов, каждый со своим объектом управления, так что позиция сток первого этапа является истоком второго и т.д. [20]. В этом случае, свойство сохраняемости получает новую интерпретацию: на каждом из фрагментов объект управления остается в единственном числе. При смене один объект управления может породить совокупность объектов другого вида, например, заявка может быть разбита на заказы по числу указанных в ней продуктов или услуг. В этом случае налицо дополнительная смена объекта управления, которую необходимо учитывать. Можно рассмотреть декомпозицию объекта управления, в этом случае модель процесса следует рассматривать как иерархию вложенных сетей Петри [21].

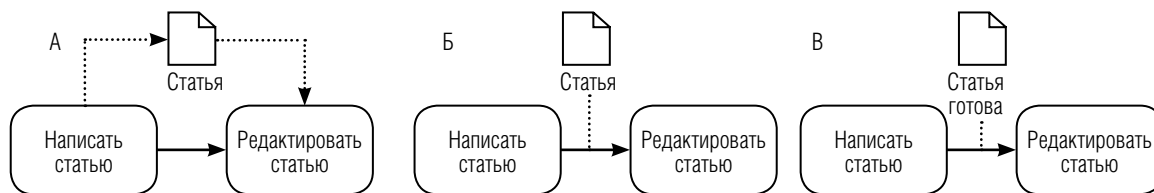


Рис. 4. Моделирование объектов данных

Сформулируем принципы моделирования процессов сетями Петри. Чтобы исключить коллизии, связанные с генерацией маркеров, надлежит рассматривать сохраняющие сети и не использовать без особого основания неструктурированные паттерны. Следует следить, с какой информационной сущностью связан каждый маркер и, в случае смены объекта управления, выделять подпроцесс, описываемый отдельной подсетью Петри. Сквозной процесс надлежит рассматривать как сеть подпроцессов, каждый из которых может быть описан WFPN.

8. Нотация BPMN

Нотация BPMN применяется для разработки исполняемой модели процесса. Она включает богатый набор элементов для описания процессов разных типов, но мы рассмотрим только их ограниченное подмножество, используемое для моделирования процессов оркестровки [10]. В качестве узлов на диаграмме выступают объекты потока управления, включающие операции, логические операторы и события. Договоримся различать узлы, где происходит изменение объекта управления, приводящее к смене его состояния и узлы, которые его не меняют. Операции трансформируют объект, его состояния на входе и выходе различаются. Логические операторы и события объект не изменяют, но маршрутизируют, первые — синхронно с потоком управления, а вторые — асинхронно. Дуги на диаграмме изображают поток управления, который выстраивает узлы в порядке их исполнения. Спецификация явно не определяет, что есть поток управления, для облегчения восприятия вводится понятие маркер, который трактуется как «теоретическая концепция», используемая для определения поведения исполняемого процесса [10]. Маркер движется вдоль модели, его текущее положение указывает на выполняемую операцию и одновременно определяет статус обработки. Попробуем установить, что есть маркер.

Выполнение бизнес-процесса всегда связано с одним или несколькими информационными по-

токами. Хотя нотация BPMN не предусматривает возможность моделирования структур данных, она позволяет изобразить информационные сущности, для этого используются графические элементы объект данных и ассоциация. Существует два способа отображения факта передачи объекта данных между операциями процесса. Во-первых (рис. 4 А), направленная ассоциация явно указывает движение информационного объекта от источника к получателю. При этом возникает ощущение, что потоки управления и данных существуют независимо друг от друга, что не верно: если поток управления направлен на логический оператор управляемый данными, то маршрутизация определяется значением информационного объекта. Во-вторых (рис. 4 Б), ненаправленная ассоциация связывает объект данных с потоком управления; такая форма подчеркивает, что потоки управления и данных суть одно и то же, что верно. Если в ходе моделирования процесса возникает желание отобразить статус обработки, можно опционально описать состояние объекта данных, оно изображается в квадратных скобках (рис. 4 В).

Процесс описывает последовательность изменения состояния системы во времени. В производственном процессе фиксируются состояния материального продукта, а в бизнес-процессе отмечаются состояния информационного объекта. Мы назвали объектом управления такой, который запоминает результат выполнения текущей операции и передает его следующей и, тем самым, связывает несколько последовательных операций процесса. Когда строится аналитическая модель процесса, в качестве объекта управления, обычно выбирается какой либо документ, важный с точки зрения выполняемого задания. Если строится исполняемая модель процесса, то выделяется переменная процесса, которая движется вдоль диаграммы. В ней происходят качественные изменения, отражающие ход выполнения процесса, ее прибытие в узел активизирует выполнение очередной операции. Переменная процесса есть суть объект управления, ее движение образует поток управления. Таким обра-

зом, маркер в исполняемой модели ассоциируется с объектом управления.

Мы ранее показали, что на некотором интервале времени состояние системы можно определить через переменную состояния. Затем мы связали объект управления процесса с переменной состояния. Теперь можно сказать, что состояние процесса определяется не только распределением маркеров по сети, но и статусом переменной состояния в текущий момент времени. Это очень важный вывод, поскольку он существенно облегчает анализ бизнес-процесса, позволяет по-новому разрабатывать его архитектуру.

Подведем итоги. Модель процесса следует трактовать как диаграмму управления, она определяет не только порядок исполнения, но временное расписание исполнения. Поток управления на диаграмме BPMN образуется в результате движения объекта управления. Смена объекта управления означает разделение сквозного процесса на подпроцессы. Сквозной процесс следует рассматривать как систему взаимодействующих подпроцессов, у каждого свой объект управления. Поскольку состояние процесса определяется не только положением маркера на диаграмме, но и состоянием самого объекта управления, диаграмма потоков работ должна быть согласована с диаграммой состояний соответствующего объекта управления. Весь процесс и отдельная операция являются сохраняющими, сумма входных потоков равна сумме выходных. Поскольку нотация BPMN не ограничивает применение неструктурированных паттернов и непарных логических операторов, устранение возникающих коллизий становится задачей аналитика.

9. Заключение

Научная новизна предлагаемой работы заключается в анализе способов формального представления динамического поведения бизнес-процесса в различных видах диаграмм. Установлено, что ни один из рассмотренных формализмов моделирования бизнес-процессов не позволяет одновременно отобразить изменение во времени всех сущностей, необходимых для описания динамики поведения системы: состояния; работы, приводящей к смене состояния; события, инициирующего выполнение работы; материального или информационного потоков, над которым выполняется работа, трансформирующая объекты, что приводит к изменению со-

стояние системы. Показано, что поток управления образуется в результате движения объекта управления, он связан с переменной состояния, характеризующей статус системы на некотором временном интервале. Удалось обосновать, что во избежание недетерминированного поведения следует использовать сохраняющие сети Петри, а предложенное ранее свойство безопасности не позволяет избежать коллизии. Сквозной процесс следует рассматривать как композицию подпроцессов, соответствующих критериям WFPN, причем в каждом из них есть только один объект управления.

Проведенный анализ показал, что модель процесса в нотации BPMN следует классифицировать не как диаграмму потоков работ, а как диаграмму потоков управления. Она изначально показывает наибольшее число объектов, характеризующих динамику поведение системы, позволяет описать расписание исполнения и бизнес-правила, имеет детализацию уровня действий, показывает все варианты исполнения.

Практическая ценность работы заключается в том, что показана взаимосвязь диаграммы потоков работ и диаграммы состояний. Статус обработки в любой момент времени определяется не только распределением маркеров по сети, но и состоянием объекта управления. Сформулирован критерий разделения сквозного процесса на подпроцессы. Для этого предлагается следить за сменой объекта управления бизнес-процесса.

Системный подход к проектированию программ предполагает последовательное моделирование в нескольких нотациях и взаимное согласование частных моделей. Используемый в этой работе метод моделирования бизнес-процесса предлагает анализ предметной области с целью выявления объектов управления и связей между ними. Предлагаются принципы, которые помогут при описании поведенческой перспективы модели бизнес-процесса:

- ◆ выделять в процессе ограниченное число объектов управления, рассматривать их как переменные состояния, определять взаимосвязи между ними;
- ◆ разделять количественные и качественные изменения переменной состояния, последние рассматриваются как комплексные;
- ◆ иерархически декомпозировать переменные состояния;
- ◆ фиксировать один объект управления на каж-

дом этапе исполнения процесса;

♦ контролировать смену объекта управления, новый объект может появиться только в результате трансформации существующего;

♦ рассматривать сквозной процесс как композицию подпроцессов, каждый со своим объектом управления;

♦ модель процесса должна удовлетворять требо-

ваниям сохранения и не допускать размножения точек управления.

Описываемые в работе принципы формального представления поведенческой перспективы модели бизнес-процесса были апробированы при создании системы управления бизнес-процессами электронного вуза, разрабатываемой в МЭСИ и показали свою высокую эффективность [22]. ■

Литература

1. Федоров И.Г. Процессно-ориентированные информационные системы // В сб.: Информационные системы и технологии; под ред. Ю.Ф.Тельнова. — М.: Юнити-Дана, 2012.
2. Axenath B., Kindler E., Rubin V. The Aspects of Business Processes: An Open and Formalism Independent Ontology. — Paderborn: Technical Report tr-ri-05-256, 2005.
3. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. — СПб: Наука, 1998.
4. Калашян А.Н., Калянов Г.Н. Структурные модели бизнеса: DFD-технологии. — М.: Финансы и статистика, 2003.
5. Zur Muehlen M., Workflow-based Process Controlling. — Berlin: Logos Verlag, 2002.
6. Böhm J. Flow diagrams. Turing machines and languages with only two formation rules // Comm. ACM. — May 1966. — 9 (5).
7. Flowcharts. Standard ECMA-4. — ECMA, 1966.
8. Software AG. Methods ARIS 7.0
9. Van der Aalst W.M.P. The Application of Petri Nets to Workflow Management // The Journal of Circuits, Systems and Computers. — 1998. — Vol. 8. — No. 1.
10. OMG. Business Process Model and Notation (BPMN). Version 2.0. — 2012. <http://www.omg.org/spec/BPMN/2.0>.
11. Буч Г., Максимчук Р. Объектно-ориентированный анализ и проектирование с примерами приложений / 3-е изд. — М.: Вильямс, 2008.
12. Дейкстра Э. Дисциплина программирования. — М.: Мир, 1978.
13. Harel D. A Visual Formalism for Complex Systems // Science of Computer Programming. — 1987. — No. 8.
14. Whitten J., Bentley L., Dittman K. Systems Analysis and Design Methods. — McGraw-Hill, 2006.
15. Yourdon E. Just Enough Structured Analysis rev. 051406. — 2006. www.yourdon.com
16. Шеер А.В. Моделирование бизнес-процессов. — М.: Весть-МетаТехнология, 2000.
17. The Machine-SUIF Control Flow Graph Library. <http://www.eecs.harvard.edu/hube/software/nci/cfg.html>
18. Федоров И.Г. Сравнительный анализ методов моделирования бизнес-процессов // Открытые системы. — 2011. — № 8.
19. Kiepuszewski B., ter Hofstede A., van der Aalst W. Fundamentals of Control Flow in Workflows // Acta Informatica. — 2002. — Vol. 39. — No. 3.
20. Питерсон Дж. Теория сетей Петри и моделирование систем. — М.: Мир, 1984.
21. Ломазова И.А. Вложенные сети Петри и моделирование распределенных систем // Труды международной конференции «Программные системы: теория и приложения», Переславль-Залесский. — М.: Физматлит, 2004.
22. Федоров И.Г., Курышев К.С., Данилов А.В. и др. Особенности проектирования системы управления бизнес-процессами на примере электронного ВУЗа // Электронное образование. — 2012. — № 4.
23. Aagesen G., Krogtstie J. Analysis and Design of Business Processes Using BPMN / Handbook on Business Process Management / Editors: J. vom Brocke, M. Rosemann. — Springer-Verlag, 2010.