

КОНЦЕПЦИЯ ПОСТРОЕНИЯ МОДЕЛЕЙ КОМПОЗИТНОГО ДОКУМЕНТООБОРОТА

Abstract: *The modern conceptual, frictional, logical and structure approach of building models of electronic document turnover systems, is which allows creation on its base of the automated information systems for different applications is discussed. As the base for building such models it is proposed to use wide approbated and well proved itself the apparatus of the theory of graphs, theory of automates. Conceptual propositions of this paper may be used for deciding theoretical problems of the electronic document turn over and creation on its base of the corresponding software.*

Key words: *electronic document-over, theory of graphs.*

Анотація: *У статті розглянуто сучасний підхід побудови концептуальної, функціональної, логічної та структурної моделей системи електронного документообігу, яка дозволяє створити на їх основі автоматизовані інформаційні системи для різного застосування. Як основу побудови таких моделей пропонується використовувати широко апробований апарат теорії графів, теорії автоматів на ін., що добре себе зарекомендував. Концептуальні положення цієї статті можуть бути використані для рішення теоретичних проблем електронного документообігу і створення на їх основі відповідного програмного забезпечення.*

Ключові слова: *електронний документообіг, теорія графів.*

Аннотация: *В статье рассмотрен современный подход к построению концептуальной, функциональной, логической и структурных моделей системы электронного документооборота, позволяющий создать на их основе автоматизированные информационные системы для различных приложений. В качестве основ построения таких моделей предлагается использовать широко апробированный и хорошо зарекомендовавший себя аппарат теории графов, теории автоматов и др. Концептуальные положения этой статьи могут быть использованы для решения теоретических проблем электронного документооборота и создания на их основе соответствующего прикладного программного обеспечения.*

Ключевые слова: *электронный документооборот, теория графов.*

1. Введение

Парадигмой применения научного подхода при решении современных задач является построение формальной модели для исследования объектов реального мира. В настоящей статье будет рассмотрена концепция построения формальной модели композитного документооборота. Композитный документооборот понимается как документооборот, который при движении документов использует как электронные, так и бумажные носители, а также их композитные решения.

В [1] понятие модели определяется следующим образом: «Модель – материальный объект или совокупность математических, логических или иных соотношений, воспроизводящих изучаемый объект в некотором отношении, заданном целью исследования». Таким образом, модель является формальной копией объекта реального мира по возможности наиболее полно повторяя его свойства и поведения. Это качество модели дает возможность использовать ее в исследованиях, подменяя ею собственно объект реального мира, что позволяет более направленно проводить исследования, акцентируя при этом внимание на интересующих исследователя свойствах.

На данный момент реализации документооборота являются композитными не только из-за используемых технологий, но и вследствие двойственности исходных установок. Выражаясь точнее, над реализациями документарной деятельности организаций довлеют постсоветские традиции, а в качестве основного инструментария используются прозападные технологии.

Вследствие этого, для построения формальной модели использовались как западные, так и отечественные теоретические наработки и практические результаты.

Из отечественной теоретической базы в качестве основы для построения модели был принят принцип автоматизации документооборота, описанный В.М. Глушковым в [2]. Из западной были использованы теоретические результаты, полученные известным американским специалистом в области документооборота Майклом Саттоном и описанные в [3].

При практической реализации представленной формальной модели использовались методы и средства дискретной математики. В отечественной и западной математических школах, несмотря на общность подходов, существуют некоторые минорные отличия, которые радикальным образом влияют на представление моделей. Отечественный подход и формы представления моделей были взяты из [4], адаптация к западным происходила после изучения работ [5] и [6].

2. Постановка проблемы

Для создания формальной модели композитного документооборота производится обследование реального объекта для выявления сценариев его поведения и протоколов свойств, совокупность которых является интерфейсом с внешним миром. Общее множество установленных протоколов и выявленных сценариев образует правила. Когда формальная модель является концептуальной, то ее поведение определяется правилами, определенными при обследовании реального объекта. При использовании любого подхода обследований из объекта реального мира вычленяется ограниченное количество признаков, специфичных для примененного метода обследования. В то же время сами объекты реального мира достаточно сложны, то есть обладают достаточно большим количеством признаков, совокупность которых представляет их восприятие. Поэтому для моделирования потребовалось бы громадное количество правил, чтобы полностью описать поведение и свойства реального объекта. При задании формальной модели используется конечное количество правил, поэтому модель всегда проще реального объекта.

Таким образом, вне зависимости от применяемой методологии обследования и инструментария задания правил модель всегда получается более простой для понимания и изучения, чем объект реального мира, но при этом утрачивается ряд свойств моделируемого объекта.

По данным, полученным при обследовании предмета реального мира, может быть сформировано конечное количество правил. На базе этих правил, в зависимости от используемого инструментария, может быть построено очень большое количество абстрактных представлений – формальных моделей. Все эти модели будут построены на основе конечного подмножества, выделенного из множества данных о его свойствах и поведении. Степень соответствия модели реальному объекту, то есть отношение подмножества использованных правил к общему множеству данных о свойстве и поведении объекта реального мира называется адекватностью модели.

В общем случае из бесконечного множества правил может быть выбрано бесконечное количество подмножеств, из которых могут быть построены формальные модели. Однако

исследователю-разработчику необходимо выбрать из множества моделей ту, в которой были бы отражены только существенные свойства изучаемого объекта.

В данной статье будет изложена методология построения формальной модели, которая позволит выбирать адекватные модели, обеспечивающие построение измеримых и управляемых систем композитного документооборота. Требование измеримости обусловлено современными требованиями к оценке прогресса внедрения информационных систем. Международная статистика успешности проектов в области информационных технологий говорит о том, что своевременное прекращение заведомо неуспешных проектов дает значительную экономию финансовых средств. Требование управляемости предопределено современными темпами изменения бизнес-задач. Интервалы успешной рыночной деятельности изменяются столь стремительно, что необходима быстрая адаптация компании при переходе из кризисных задач к ежедневным. Возможность быстро и эффективно изменять используемые бизнес-процессы обеспечивает компании поддержку в непрерывной адаптации к стремительно меняющейся конъюнктуре рынка.

3. Методология построения формальной модели

При создании программного обеспечения, которое адекватным образом учитывает требования всех участников документооборота и позволяет адаптивно реагировать на внесенные изменения, целесообразно использовать метод построения программного обеспечения на основе формальной модели.

Данный метод является одним из вариантов широко используемого сегодня прецедентного подхода, который используется при создании современного программного обеспечения. Появившись вместе с объектно-ориентированным подходом, прецедентное программирование развилось и установилось, являясь на данный момент одним из самых мощных и отлаженных инструментариев для распределенного программирования. Подробнее с этим методом, его применением и технологией IBM Rational Unified Process можно ознакомиться в работе [7].

Суть этого метода состоит в том, что создание программного обеспечения, реализующего отображения некоторого реального объекта происходит через создание нескольких моделей. Весь процесс создания программного обеспечения приводится к рационально унифицированному процессу, который укрупненно разбивается на три основных этапа: теоретический, практический и физический. Каждый из этапов характеризуется получением промежуточного результата, который является входными данными для следующего этапа. Результаты – формализованные, а процесс получения результатов – измеримый и контролируемый. Кратко описать процесс можно следующим образом: сначала происходит построение общей абстрактной модели, которая потом, путем преобразования общей модели, превращается в конкретную модель. На основании полученной таким образом конкретной модели производится ее реализация в отображение, то есть собственно создание программного обеспечения для конкретного применения. Перечислим особенности каждого из вышеназванных этапов рационально унифицированного процесса.

Первый этап – построение общей теоретической модели. На этом этапе производится сбор данных об объекте реального мира и создание теоретического описания. Описание строится таким

способом, чтобы максимально полно покрывать свойства и поведение моделируемого объекта. Как уже отмечалось выше, сколь долго бы не велись исследования, сколько полно использовалась бы информация, все равно невозможно достичь полного соответствия теоретической модели реальному предмету. Этот очевидный факт естественным образом следует из факта бесконечности множеств, свойства и поведения реального объекта. Поэтому общая модель строится таким образом, чтобы быть покрывающей. То есть настолько общей, чтобы избыточно покрывать информацию, полученную при исследовании реального объекта. Эту характеристику покрываемости можно сформулировать следующим образом: общая модель должна быть избыточной и достаточной. Это следует из того, что могут существовать объекты из реального мира, не относящиеся к объекту, которые подпадают под описания и присутствуют в модели. При этом у объекта не существует свойств, которые бы не были описаны, поведения, которое не было бы запротоколировано.

Второй этап – построение конкретной теоретической модели объекта реального мира. В качестве входных данных используется полученная на предыдущем этапе общая теоретическая модель. При этом общая теоретическая модель является избыточной, и ее описание часто покрывает не только моделируемый объект, но и некоторое количество объектов, имеющих сходные свойства или похожее поведение. Конкретная теоретическая модель строится путем применения аппарата существующей теории к общей модели. При этом происходит ее адаптация к уже установившимся формулировкам, понятиям и используемому инструментарию. В процессе такой адаптации, в силу ограниченности любой существующей теории, модель может утратить некоторые свойства, которые не описывают применяемый аппарат или не реализуют используемый инструментарий. Таким образом, конкретная модель получается достаточно формализованной и может быть практически реализована имеющимся инструментарием. Однако при этом происходит снижение адекватности в силу ограниченности используемого аппарата существующей теории, что приводит к утере свойств моделируемого объекта. Тем не менее, конкретная теоретическая модель позволяет создавать практическую реализацию объекта, то есть реализовать конкретную задачу моделирования.

Третий этап – построение практической (физической) реализации объекта реального мира. Входными данными для этого этапа является полученная на предыдущем этапе конкретная теоретическая модель. Это модель характерна тем, что описана в терминах и правилах конкретной существующей теории и для ее реализации подготовлен существующий инструментарий. Этот этап является собственно программированием, созданием прикладного программного обеспечения на основе полученных при исследовании описаний, которые основываются на апробированных средствах разработки. Результатом данного этапа является некоторое программное обеспечение, которое само по себе тоже является объектом реального мира. Этот результат есть экзemplаром, полученным на основе отображения моделируемого объекта с использованием аппарата теорий реального мира.

Использование вышеизложенной этапности позволяет разбить инновационный процесс создания программного обеспечения на конечный список определенных работ. Значительным достоинством этой методологии является возможность проведения проверок промежуточных

результатов работ – моделей, получаемых на первом и втором этапах. Такая промежуточная проверка позволяет потребителю программного обеспечить прозрачность и управляемость процесса разработки. Разработчик программного обеспечения с помощью этой методологии решает задачу изолированности своих производственных сил – программистов от реальных задач и проблем потребителя. Потребитель по факту анализа полученных промежуточных моделей может скорректировать планы разработчиков, что дает значительную экономию финансовых средств и времени. Более того, связность получаемых моделей позволяет принимать решения и производить изменения в системе на уровне представлений общей архитектуры системы, что дает возможность абстрагироваться от текущих проблем реализации, уделить больше внимания значимым сущностям и мыслить на системном уровне. Это позволяет распределить компетенцию команды, которая занята созданием программного обеспечения и снижает конфликты между специалистами прикладной области, в интересах которых решается данная задача, и компьютерными специалистами, которые реализуют задачу.

Вышеизложенная методология схематично отображена на рис 1.

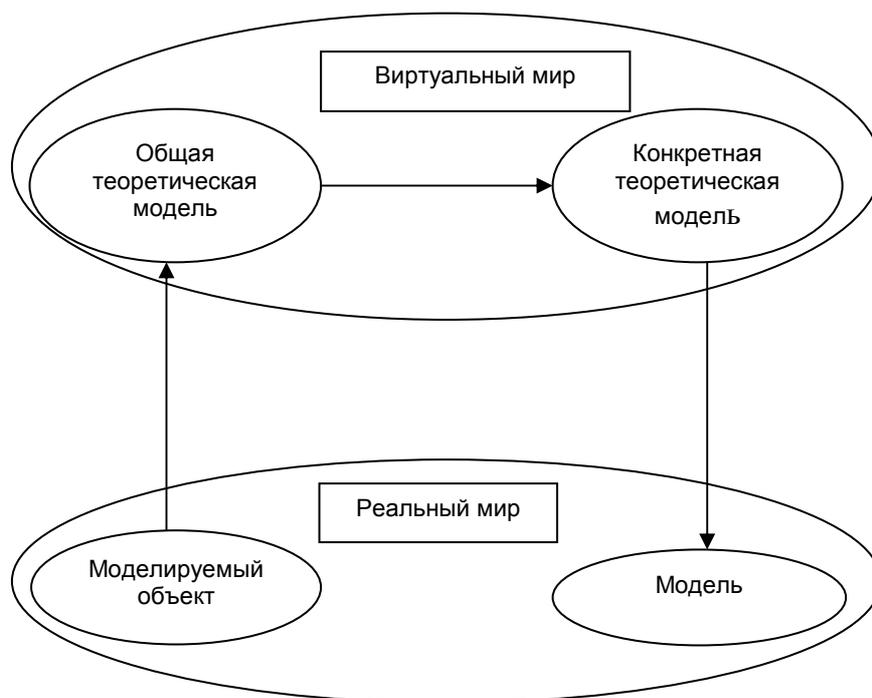


Рис. 1. Методология создания системы документооборота

На рис. 1 в общий контур объединены объект и его программная реализация, общая теоретическая модель и конкретная теоретическая модель. Это сделано для того, чтобы показать существующую общность между ними. При этом имеется в виду, что объект и реализация находятся в реальном мире, то есть имеют некоторое физическое воплощение. В отличие от этого, обе модели находятся в виртуальном мире, то есть их воплощением является некоторый набор теоретических сведений. Таким образом, видно, что создание программной реализации объекта реального мира предполагает использование его теоретических проекций.

Если принять вышеуказанное обобщение, то можно говорить о двух этапах проводимых преобразований: из реального мира в теоретический и из теоретического в реальный. По применяемому инструментарию эти этапы можно назвать абстрагированием и реализацией. Этап абстрагирования основан на обобщении проецирования объектов реального мира в теоретический, то есть их воссоздание через систему свойств и признаков. На этапе реализации происходит обратный процесс: по предварительно заданным теоретическим данным функционально воссоздается объект реального мира.

При этом абстрагирование понимается в аспекте анализа предмета с целью выявления его сущностей, выработки критериев значимости, выявление значимых сущностей, а также отсева незначимых. Степень значимости сущности определяется степенью ее отношения к изучаемым свойствам объекта. Таким образом, абстрагирование, с одной стороны, дает возможность концентрировать внимание разработчиков на исследуемых свойствах объекта, а, с другой стороны, за счет отсева незначимых сущностей понижает уровень адекватности исходного объекта и полученной модели.

В свою очередь, реализация позволяет воссоздание модели по некоторому формализованному описанию свойств и поведений используя существующий теоретический аппарат и прикладной инструментарий. Как следствие этого, модель подвержена влиянию технологических особенностей использованного аппарата и существующих ограничений применяемого инструментария. Таким образом, любая реализация является источником снижения адекватности модели и полученного из него объекта реального мира.

Как видно из вышеизложенного, в обоих случаях возникают проблемы адекватности. Причем, проблема адекватности возникает как при получении общей теоретической модели из предмета реального мира, так и при синтезе программной реализации из конкретной теоретической модели.

Вышеописанные случаи можно рассматривать также и с точки зрения используемых средств: декомпозиции и синтеза. При абстрагировании происходит разделение рассматриваемого предмета на множество значимых и незначимых сущностей. Таким образом, при помощи декомпозиции достигается представление целостной системы в виде совокупности подсистем. При реализации происходит обратный процесс: из множества отобранных и формализованных сущностей синтезируется объект, свойства и поведение которого подобны свойствам и поведению исходного объекта.

Глубина проводимой декомпозиции определяется особенностями каждой конкретной реализации и зависит от требований, предъявляемых к системе, и ресурсов, выделенных для ее реализации. Чем глубже декомпозиция, тем меньше размер элементарных единиц систем и тем точнее уровень воспроизведения. В то же время, чем больше глубина, тем больше количество элементов, сложнее отношения между ними, тем сложнее последующая реализация самой системы. Декомпозицию принято делать до такой глубины, чтобы можно было обеспечить измеримость полученных составных элементов без потери управляемости процессов.

В настоящей статье предлагается методология, которая предполагает проведение декомпозиции процесса документооборота до совокупности элементов и их отношений между

собой. Эти элементы можно разделить на три категории: участники, состояния документов и действия участников. Участники документооборота – это сотрудники организации, производящие генерирование, движение и терминирование документов. По сути, участники воспринимаются системой через совокупность их обязанностей и могут описываться как ключевые участники, на роли которых назначаются реальные исполнители. Состояния документов – это конечный список состояний, которые могут принимать документы в процессе моделируемого документооборота. Представление документов в виде конечного дискретного списка состояния получается в результате применения декомпозиции к общему жизненному циклу документа. Действия участников – это конечный список возмущающих воздействий, инициируемых участниками, возникновение которых приводит к изменению текущего состояния одного или нескольких документов.

Таким образом, формально процесс документооборота может быть представлен в виде трех конечных множеств и связей элементов этих множеств между собой. Математическая нотация этого процесса может быть представлена в виде тройки $D_T = \{U, D, \Phi\}$, где

D_T – формальная модель документооборота;

U – множество участников;

D – множество действий;

Φ – множество состояний документов.

Эта нотация означает следующее: «Документооборот – это множество действий, производимых множеством участников над множеством состояний документов». Множество U определяется как конечное множество ролей, которые могут быть назначены фактическим участникам документооборота. D определяется как конечное множество действий, выполнение которых допустимо в пределах рассматриваемой системы документооборота. Φ – конечное множество состояний, которые могут принимать документы после произведения действий из множества D участником из множества U .

3.1. Концептуальная модель

Концептуальная модель основывается на подходах, описанных в работах признанного специалиста в области создания систем корпоративного документооборота Майкл Саттона [3], В.М. Глушкова [2] и автора этой статьи [8]. В концептуальной модели на уровне концепции решаются вопросы масштабируемости системы и ее интеграции в общую систему организации. Для создания модели устанавливается взаимосвязь между необходимостью внедрения системы электронного документооборота и ее будущими пользователями.

Концептуальная модель разрабатывается таким образом, чтобы руководство организации могло наглядно представить себе будущую систему в общем виде и через это понимание была обеспечена поддержка реализации системы высшим руководством организации. Создание и внедрение информационной системы по сути является расширением существующей системы управления организации возможностями информационных технологий. Поэтому поддержка и

понимание первого руководителя организации является критичным критерием для успешности внедрения информационной системы. В.М. Глушков назвал такой подход принципом первого руководителя и не просто упоминает о нем, а акцентирует внимание на важности обеспечения этого фактора.

Для построения концептуальных моделей согласно вышеописанным принципам надо получить данные, которые смогут достаточно адекватно описать моделируемый процесс. Для этого целесообразно общий процесс декомпозировать на совокупность элементов нижнего уровня согласно заданной выше нотации. Общий процесс декомпозируется на три основные части: участники, действия и состояния. Эти части, каждая из которых является целостным технологическим элементом системы, далее декомпозируются на элементарные составляющие. Полученные элементы группируются в множества $У$, $Д$ и $Ф$.

Для получения множества участников $У$ используются данные, полученные на этапе анализа системы документооборота [9]. На этом этапе выявляются характерные повторяющиеся участки, свойственные для установившихся ролей. Для этой общности строится список ролей. На основании списка ролей определяются ключевые участники, которые могут быть назначены для выполнения описанных ролей.

Критерием успешности проведенной декомпозиции являются полнота и невырожденность множества $У$. То есть декомпозиция может быть проведена с избыточностью таким образом, чтобы одному физическому участнику соответствовало несколько ролей. Допустима ситуация, в которой одному и тому же действию в реальной жизни может соответствовать несколько действий формализованных ролевых персон. В то же время недопустимо вырождение множества $У$, то есть ситуация, в которой физическому участнику не установлено никакой роли.

Множество состояний $Ф$ получается путем составления конечного списка состояний, допустимых для документов, обращающихся в данном документообороте. По сути происходит дискретизация жизненного цикла документа. Документ, который изменяется и движется в реальном времени, представляется в виде совокупности дискретных состояний. Каждое такое состояние характеризуется формализуемостью формы, то есть состояние может быть представлено в виде конечного количества полей и реквизитов документа. Состояния являются дискретными, конечными и описуемыми.

Множество действий $Д$ получается путем декомпозиции действий, производимых в реальной системе документооборота, на конечную совокупность элементарных действий. Каждое такое действие влечет за собой изменение состояния одного или нескольких документов из множества $Ф$. Возможен случай, когда в результате действия происходит изменение состояния само на себя. В таких случаях говорят о цикличности процесса. Несмотря на то, что цикличная организация весьма опасна с точки зрения окончания процесса, этот способ очень широко присутствует в реальных системах документооборота, так как позволяет прозрачно организовать бесконечные процедуры, имеющие выход по заданному критерию.

3.2. Функциональная модель

Функциональная модель документооборота – это описание модели системы на языке выполняемых ею функций. Электронный документооборот как любая задача информационных технологий является вторичной по отношению к автоматизируемому объекту. Реализация основывается на наличии некоторого исходного объекта, обладающего определенными свойствами и интерфейсом взаимодействия с внешним миром. Исходным объектом систем электронного документооборота является документооборот, реализующийся организацией в реальном мире. Процессы этого документооборота имеют на своем входе некоторые исходные документы и по выполнении критериев окончания генерируют на выходе конечные документы. Таким образом, упрощенно документооборот можно представить как некий инструментарий, обеспечивающий движение документов от исходного состояния к конечному.

Возвращаясь к заданной в разд. 3 нотации, можно сказать, что в множестве Φ представлены некоторые состояния, которые имеют специальные свойства, определяющие эти состояния как конечные. Состояния, обладающие таким свойством, будем называть конечными состояниями. По достижении конечного состояния, процесс, которые реализует переход в данное состояние, считается окончившимся. Кроме конечных состояний, существуют еще и начальные, которые могут быть как некоторым состоянием из множества Φ , так и пустым состоянием. Маршрут движения документа – последовательность действий, которые происходят в рамках процесса документооборота при достижении документов конечного состояния из начального. Таким образом, функциональная модель документооборота может быть представлена в виде совокупности начальных состояний, связанных с конечными состояниями маршрутами движения. Это можно наглядно отобразить в виде детерминированного или недетерминированного конечного автомата.

Рассмотрим данные, которыми оперируют системы документооборота вне зависимости от прикладной области и способа реализации. Входные данные – информация, которая поступает в систему – некоторый начальный набор состояний, возникновение которых указывает на начало работы системы. Окончательные результаты – данные, которые получаются в результате обработки системой входных данных – набор состояний, при достижении которых система принимает решение об окончании работы. Промежуточные результаты – результаты переработки исходных данных, которые используются при получении окончательных результатов, но сами из системы не выдаются. По сути, промежуточный результат – набор состояний, которые входят в общее множество состояний, но не являются ни начальными, ни конечными состояниями. Все эти состояния имеют одну значимую общность – исходные данные. Окончательные и промежуточные результаты могут быть описаны в форме слов в алфавите системы.

При рассмотрении функциональной модели документооборота можно говорить об исходных данных, промежуточных и окончательных результатах как об элементах информационного потока. Эти элементы можно перенумеровать и обозначить через $X_i (i = 1, 2, \dots, n)$. Общая совокупность всех элементов информационного потока составляет информационный базис системы. Информационный базис не зависит от программ переработки информации, а определяется, в основном, внешними функциями системы.

Между элементами потока существуют отношение вхождения и отношение порядка. Отношение вхождения имеет вид строки $X_i = X_{i1}, X_{i2}, \dots, X_{in}$, которая означает, что элемент, записанный слева от знака равенства, образуется непосредственно из элементов входа, записанных справа. Отношение порядка позволяет различать такты в движении потока. Исходные данные являются элементами нулевого порядка. На первом такте из исходных данных образуются элементы первого порядка. На втором такте из элементов нулевого и первого порядков образуются элементы второго порядка и т.д. Таким образом, порядок P_i элемента $X_i (X_i = X_{i1}, X_{i2}, \dots, X_{in})$ на единицу больше максимального из порядков элементов $X_{i1}, X_{i2}, \dots, X_{in}$.

Таким образом, мы имеем возможность представления моделируемого документооборота в виде последовательности дискретных событий. Общая совокупность этих событий состоит из конечного множества состояний. Состояниям могут быть присвоены признаки начальных, конечных или промежуточных результатов. Изменение состояний имеет детерминированную последовательность, которая может быть представлена в виде набора функций перехода. Приведенное выше описание позволяет сделать вывод о возможности представления систем документооборота детерминированным конечным автоматом.

В настоящей статье предлагается представлять моделируемую систему документооборота в виде детерминированного конечного автомата, заданного в виде нотации, описанной в [9]. Исходя из этой нотации, автомат, моделирующий документооборот, может быть представлен следующим образом:

$$D_T = (Q, \Sigma, \delta, q_0, F),$$

где Q – конечное множество состояний, тождественное множеству Φ из нотации, используемой в настоящей статье для представления документооборота;

Σ – конечное множество входных символов, образующих входной алфавит и представляющее собой данные, которые поступают на вход системы документооборота;

δ – функция переходов, аргументами которой являются текущее состояние и входной символ, а значением – новое состояние;

q_0 – начальное состояние (или множество начальных состояний) из множества Q ;

F – множество заключительных, или допускающих состояний из множества Q .

Представление формальной модели документооборота с помощью автоматов представляется самостоятельным интересом и может явиться предметом самостоятельного исследования.

3.3. Логическая модель

После актуализации тройки множеств V, D, Φ , описывающих формальную модель системы документооборота, и построения таким образом концептуальной модели появляется возможность построить логическую модель. По Майклу Саттону [3] логическая модель должна дать ответ на

вопросы «Что» и «Когда». Исходя из пользовательской схемы документооборота и протоколов взаимодействия элементов системы, определяется: «Что будет делать система?» и «Когда должен запускаться каждый из процессов?»

На логическом уровне решаются вопросы функциональных характеристик системы электронного документооборота типа ввод и вывод данных, обработка данных, протоколы политики безопасности, правила ведения дел, составление форм, а также форма и периодичность отчетов и т.п. Логика построения системы электронного документооборота на этом этапе не имеет привязки к конкретной системе, на которой она будет запущена. Установленное логическое построение может быть реализовано при различной аппаратной и программной реализации.

Возвращаясь к заданной в разделе 3 нотации, можно сказать, что на уровне реализации логической модели выделяются и однозначно устанавливаются связи, определяющие зависимость состояний из множества Φ . Логика документооборота представляется в виде последовательности действий, приводящих к смене состояний документов в системе документооборота. Таким образом, формируется логически связанная последовательность действий, преобразующая документ от начального состояния к требуемому - конечному.

Логическую модель наглядно можно представить в виде направленного плоского геометрического графа. Для установления соответствия графическому отображению введенной в данной статье нотации документооборота может быть использована так называемая парная грамматика. Парная грамматика представляет собой композицию двух грамматик, между правилами и нетерминальными символами, между которыми устанавливаются определенные соответствия. Таким образом, парная грамматика устанавливает связь между элементами языков, определенных двумя грамматиками. Эта связь может рассматриваться как определение перевода элементов одного языка в другой. В нашем случае рассматривается вариант, в котором первый язык – тройка множеств U, D, Φ , а второй – набор графов с помеченными дугами и вершинами.

При построении графовой модели документооборота предлагается использовать следующий способ отображения документооборота: множество возможных состояний используется для обозначения вершин графа, а множество действий - для обозначения ребер графа. Используя нотацию, принятую для математического представления графа можно сказать, что $v(i) = \Phi(i)$ и $e(i) = D(i)$. Направленность ребер отражает логику последовательности смены состояний.

Таким образом, состояниям y_1, y_2, \dots, y_n сопоставляются вершины графа v_1, v_2, \dots, v_n , и каждая пара вершин v_i и v_j соединена дугой, идущей от v_i к v_j в том и только том случае, когда состояние v_i является входным состоянием для v_j . Полученный граф будем рассматривать как логическую модель документооборота.

Пример такого графа приведен на рис. 2.

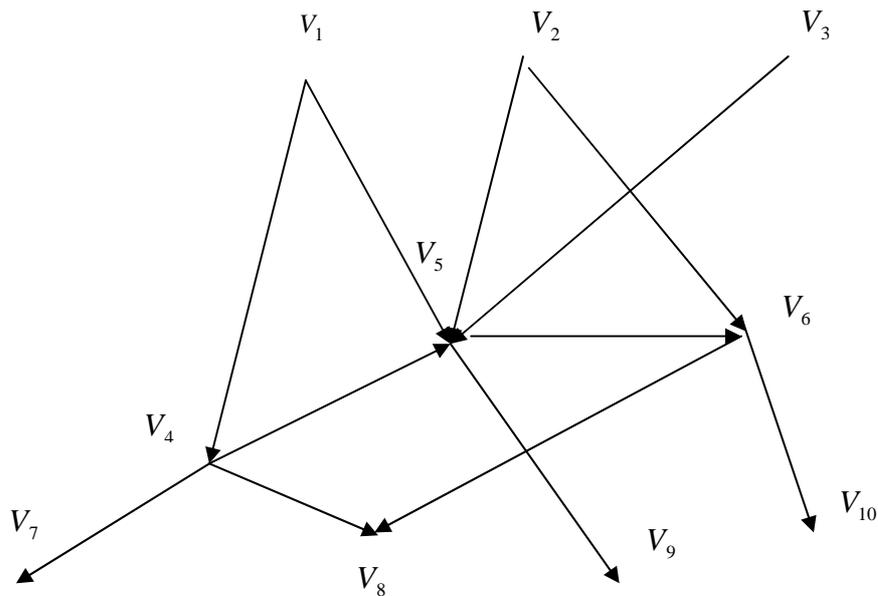


Рис. 2. Фрагмент графовой модели системы документооборота

Пример расширенной логической модели приведен на рис. 3.

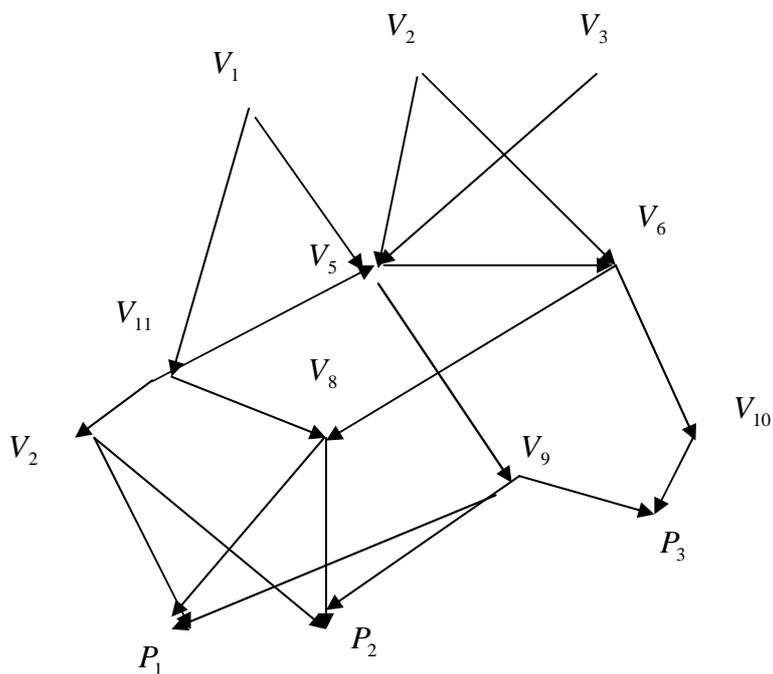


Рис. 3. Фрагмент расширения логической графовой модели системы документооборота

Схему можно усложнить, введя в нее вершины P_j , соответствующие участникам системы документооборота, то есть элементам множества \mathcal{U} . Если участник работает с состоянием v_i , то

v_i является входом для P_j . Из указанной вершины графа v_i проводится дуга, которая оканчивается в вершине P_j . Таким образом, получается граф, состоящий из вершин v_i и P_j и ориентированных связей между ними. Отметим, что в этом графе нет дуг, выходящих из P_j . Такой граф будем называть расширенной логической моделью документооборота.

Пользуясь известными свойствами графов, можно выявить ряд важных характеристик логических моделей документооборота. Приведенные ниже результаты будут получены с помощью применения базовых положений аппарата теории графов.

Для математического задания графа воспользуемся его представлением с помощью матрицы инцидентности. Пусть задан граф $G(V, E, \Delta)$, где V – непустое множество, E – множество не пересекающееся с V , Δ – отображение множества E на $V \times V$. Элементы V и E соответственно называются вершинами и дугами, а Δ называется ориентированным отображением инцидентности ориентированного графа. Матрицей инцидентности ориентированного графа $G(V, E, \Delta)$, содержащего n вершин, называется квадратная матрица A с n строками и n столбцами, в которой элементы E_{ij} , стоящие на пересечении i -й строки и j -го столбца, численно равны количеству дуг графа, идущих из i -й вершины в j -ю вершину.

Ориентированным маршрутом (путем) длины n является последовательность (не обязательно различных) дуг e_1, e_2, \dots, e_n таких, что соответствующая им последовательность вершин формирует направленный граф. В нашей модели путь означает процесс создания документа от какого-то начального состояния до состояния, которое принимается конечным в правилах создаваемой модели.

Кроме того, если обозначить $G(V, E, \Delta)$ – как граф моделируемого документооборота, а A – матрицей инцидентности заданного графа, можно утверждать, что:

1) элемент $a(\lambda)_{ij}$ матрицы A^λ , полученный возведением матрицы A в степень λ , равен числу различных путей длины λ , идущих от v_i к v_j ;

2) признаком контура (ошибка описания) служит появление ненулевых элементов на главной диагонали любой из матриц A^λ ;

3) равенство нулю суммы элементов j -го столбца матрицы инцидентности служит формальным признаком для выделения исходных данных;

4) равенство нулю суммы элементов i -й строки указывает на наличие функциональных результатов;

5) если при некотором $i = j$ нулю одновременно равняются суммы элементов строки и столбца, то указанное состояние не используется в модели описываемого документооборота.

Отношение вхождения «состояние-участник» графа расширенной логической модели установлено только для активных состояний. Отличные от нуля элементы матрицы A , где P_j – участник, указывают все активные состояния моделируемого документооборота. Ненулевые элементы тех же столбцов матрицы указывают как активные, так и пассивные состояния,

используемые при достижении активных состояний. Остальные состояния, введенные в модель, формально являются избыточными, и их исключение не сможет ни позитивно, ни негативно сказаться на адекватности модели.

Вышеописанные результаты получены непосредственно из базовых понятий теории графов, а само отображение модели документооборота с использованием аппарата теории графов представляет особый интерес и требует дополнительных исследований.

3.4. Структурная модель

Структурная модель реализует представление системы в виде четко выраженных структурных единиц, различающихся по организации и выполняемым задачам. При всем множестве задач, которые ставятся перед системами документооборота предприятий, можно провести разделение, выделив задачи, решающие общие проблемы подобными методами. Таким образом, можно получить компоненты, реализация которых сделает систему достаточной, то есть обеспечит выполнение необходимых требований.

С точки зрения решаемых системой задач можно выделить две основные структурные компоненты: модуль для проектирования бизнес-процессов и модуль, реализующий спроектированные процессы в деятельности предприятия. Обе компоненты используют одно общее хранилище, которое содержит образы предопределенных и экземпляры активных в данный момент бизнес-процессов. Майкл Саттон в [3] называет такое хранилище репозитарием и акцентирует внимание на концентрации усилий разработчиков и внедренцев систем ЭД на продуманные реализации.

Создание активных копий бизнес-процессов из образов, хранимых в библиотеке предопределенных процессов, позволяет организовать создание системы документооборота в соответствии с канонами объектно-ориентированного анализа и проектирования, описанными в [10]. Непротиворечивость принципов объектно-ориентированного анализа и проектирования методологиям создания систем композитного документооборота уже рассматривалась автором настоящей статьи и подробнее с аспектами этого вопроса можно ознакомиться в [9].

Таким образом, можно представить достаточные условия создания системы как два структурных модуля. Назовем их «Архитектор» и «Двигатель». «Архитектор» обеспечивает размещение в репозитарий троек вида $\{Y, D, \Phi\}$ и поддержку процессов поддержания этих троек в состоянии, актуально отражающем деловые процессы предприятия. «Двигатель» обеспечивает использование репозитария для получения образов из библиотеки процессов и по этим образам – создание активных экземпляров, реализующих документарную деятельность предприятия.

3.5. Управленческая модель

Управленческая модель подразумевает представление документооборота в виде множества взаимодействующих процессов с установленными между ними связями. В этом случае процесс представляет собой совокупность действий, объединенных некоторой общностью. Представление

общего процесса деятельности предприятия в виде совокупности процессов имеет ряд преимуществ. Например, позволяет руководству более наглядно представить деятельность своей организации, а подчиненным – представить выполняемую работу более понятной.

Хочется отметить, что управление процессами является одной из современных апробированных форм управления и широко используется во всем мире. Декомпозиция общей деятельности на совокупность процессов позволяет представить каждый процесс в формализованном виде, а именно – снабдить процессы четким описанием, описать входные и выходные параметры. Представленные таким образом процессы обладают свойствами управляемости и контролируемости. Это имеет целый ряд положительных последствий. Например, позволяет собственникам более эффективно управлять предприятием и контролировать нанятых высших управленцев. Кроме того, на управлении процессами управления основаны современные методологии систем управления качеством, предусмотренные ISO9000.

Математическое представление процесса в рамках введенной нотации можно представить как подмножества из множества тройки $\{U, D, \Phi\}$. Совокупность таких подмножеств представляет собой общую деятельность предприятия и является одной из форм представления, которая может быть получена в рамках формальной модели документооборота.

4. Выводы

В статье рассмотрен современный подход к построению концептуальной, функциональной, логической и структурных моделей системы электронного документооборота, позволяющий создать на их основе автоматизированные информационные системы для различных приложений.

В качестве основ построения таких моделей предлагается использовать широко апробированный и хорошо зарекомендовавший себя аппарат теории графов, теории автоматов и др. Концептуальные положения этой статьи могут быть использованы для решения теоретических проблем электронного документооборота и создания на их основе соответствующего прикладного программного обеспечения.

СПИСОК ЛИТЕРАТУРЫ

1. Справочник-словарь терминов АСУ / В.И. Вьюн, А.А. Кобозев, Т.А. Паничевская, Г.С. Теслер. – М.: Радио и связь, 1990. – 128 с.
2. Глушков В.М. Введение в АСУ. – К.: Техніка, 1972. – 312 с.
3. Саттон М.Дж. Корпоративный документооборот. – М.: Азбука, 2002. – 448 с.
4. Основы дискретной математики / Капітонова Ю.В., Кривий С.Л., Летичевський О.А., Луцький Г.М., Печурін М.К. – К.: Наукова думка, 2002. – 578 с.
5. Peter Grossman. Discrete mathematics for computing. – London: Macmillan Press, 1995. – 290 p.
6. Anderson J.A. Discrete mathematics with combinatorics. – New Jersey: Prentice Hall, 2001. – 807 p.
7. Booch G., Rumbaugh, Jacobson The Unified Modeling Language. – NY.: Addison-Wesley, 1999 – 482 p.
8. Круковский М.Ю. Методология построения композитных систем документооборота // Математичні машини і системи. – 2004. – № 1. – С. 101 – 114.
9. Booch G. Object-oriented analysis and design. Second edition. – NY.: The Benjamin, 1994. – 589 p.
10. Хопкрофт Д., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений. – М.: Издательский дом «Вильямс», 2002. – 528 с.