

## МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ В ЦИФРОВУЮ ЭПОХУ

**А. Н. ТЕРЕХОВ, М. В. ПЛАТОНОВА**

*Математико-механический факультет,  
Санкт-Петербургский государственный университет, Россия*

Тренд цифровизации только несколько лет назад появился в российском обществе и уже активно внедряется практически во все его сферы. Моделирование бизнес-процессов с помощью современных ИТ-подходов является важнейшим резервом повышения эффективности и надежности производственных комплексов. К сожалению, редки исследования, предлагающие конкретные средства формального описания производственных процессов. Авторы представляют четырехэтапную систему моделирования бизнес-процессов, включающую: конструирование, визуализацию, построение логической модели (переведение графической модели в логическую модель, на базе которой строится математическая модель, как правило, основанная на применении теории графов) и, наконец, автоматизацию генерации компьютерных кодов (CASE-технология). Данная технология моделирования бизнес-процессов и описываемый конкретный продукт могут найти широкое применение в процессах производства, начиная от автоматизации документооборота и заканчивая работой станков с числовым программным управлением. Предложенный инструмент может способствовать увеличению продуктивности персонала организаций и оптимизации производственных процессов; автоматическая генерация избавляет от основного недостатка прикладного программного обеспечения, так как код не содержит ошибок.

*Ключевые слова:* цифровизация, визуализация, бизнес-процесс, документооборот, числовое программное управление, стандарт BPMN, CASE-технология.

*JEL:* M53.

Прогресс в сфере информационных технологий (ИТ) оказал огромное влияние на развитие экономики [Лонский, 2015]. Се-

годня ИТ вовсе не являются вспомогательным средством, а составляют базу активных производственных процессов. История

Адрес организации: Санкт-Петербургский государственный университет, Математико-механический факультет, Россия, 198504, Санкт-Петербург, Университетский пр., 28.

© А. Н. Терехов, М. В. Платонова, 2019

<https://doi.org/10.21638/spbu18.2019.404>

создания автоматических систем управления (АСУ) насчитывает уже много десятилетий, но реальные успехи не столь многочисленны. На наш взгляд, основная часть проблем была связана с недостаточным уровнем формализации производственных процессов, что давало большой простор волюнтаристским решениям. Существует необходимость в создании средств формального описания производственных процессов, выраженных в цифровой, т. е. доступной компьютерам, форме.

Россия относительно недавно встала на путь цифровизации<sup>1</sup>, при этом демонстрирует хорошие темпы развития и даже имеет перспективы занять лидирующие позиции в рейтинге цифровых экономик мира. В связи с этим нарастает зависимость российской экономики от использования качественного программного обеспечения. Требуется наличие собственных программных продуктов, конкурентоспособных на международном уровне. В настоящее время большая часть государственных учреждений перешла на отечественные информационные системы, тогда как в отраслях производства процент внедрения российских разработок стремится к нулю.

Создание отдельных программных продуктов не решает задачу — необходимо сформировать целую платформу средств, относящихся ко всем стадиям производственного цикла. Например, нужны графические средства проектирования и конвертации графических моделей в программы на алгоритмическом языке высокого уровня, трансляторы проектов в коды для различных аппаратных платформ, отладчики, средства статического и динамиче-

ского контроля и т. д. Открытие успешного бизнеса на базе подобной платформы — довольно непростая задача, предполагающая налаживание взаимосвязей между различными архитектурными решениями. В данной статье приведено описание платформенного решения, а также примеры его реализации для различных производственных задач.

В рамках утвержденной российским правительством программы цифровой экономики поставлено множество различных задач, среди них — автоматизация производства [Плотников, 2018]. В данной области выделяют различные направления деятельности, к примеру, графическое представление производственных процессов, что подразумевает описание некой совокупности действий в виде одной или нескольких диаграмм [Alotaibi, 2017]. Для этих целей популярно использование международного стандарта BPMN (Business Process Model and Notation).

У диаграммного представления производственных процессов есть и другая крайность — «картинка для начальника». На наш взгляд, визуализация не только должна решать задачу наглядности, но и служить средством формализованного контроля корректности процессов, в том числе проверки их исполнения, т. е. стать полноценной платформой проектирования с использованием диаграмм. Такие диаграммы доступны для понимания всем заинтересованными лицам и упрощают дальнейшую работу с процессами, так как именно визуализация способствует улучшению понимания и осмысления необходимых сведений [Шеер, 2000; Полякова, 2012; Афанасьев, 2014]. При этом средства визуализации процессов способны обеспечить возможность симуляции и автоматической генерации кода исполняемой программы [Vasilecas, Kalibatiene, Lavbič, 2016]. Соответственно, создание решения для визуализации процессов с возможностью последующей генерации кода является акту-

<sup>1</sup> См., напр.: Распоряжение Правительства Российской Федерации от 28 июля 2017 г. № 1632-р «Цифровая экономика Российской Федерации»; Указ Президента Российской Федерации от 9 мая 2017 г. № 203 «О стратегии развития информационного общества в Российской Федерации на 2017–2030 годы».

альной задачей не только для области ИТ, но и для экономической и управленческой сфер деятельности.

Каждому типу процессов соответствуют определенные генераторы, поэтому важно учитывать особенности описываемых процессов и требования к совместимости кодов разных систем. Кроме того, генератор может накладывать некие ограничения на диаграмму, например, на создание, модификацию или удаление ее объектов либо на переходы между ее состояниями. Следует иметь в виду, что продукт изготавливается для людей, не всегда имеющих техническое образование, поэтому рабочий процесс должен быть максимально простым и удобным.

Особенности операционных систем могут значительно повлиять на разработку платформенного решения, так как не все системы обладают необходимыми свойствами и реализация технологий под каждую из них в отдельности значительно замедляет и усложняет процесс. Таким образом, использование среды разработки, совместимой со многими системами, не только упростит реализацию ИТ-решений, но и расширит возможности их применения.

Данная работа посвящена разработке кроссплатформенного приложения для визуализации производственных процессов и генерации соответствующего им кода.

На кафедре системного программирования математико-механического факультета СПбГУ уже проводилась схожая работа по созданию средства для графического описания бизнес-процессов с возможностью автоматической генерации кода. Основываясь на предыдущем опыте, можно отметить, что использование малораспространенного, сложного или имеющего много ограничений инструментария может привести к проблемам с развитием и поддержкой продукта. Нами используется популярная среда разработки Eclipse — бес-

платная кроссплатформенная система, постоянно развивающаяся и охватывающая большое сообщество разработчиков, что облегчает решение возникающих проблем. Кроме того, важно учитывать различия между существующими лицензиями при использовании уже готовых продуктов. Например, General Public License (GPL) подразумевает сохранение статуса open-source для нового продукта, т.е. не разрешает создание на его основе коммерческих продуктов. Вместе с тем лицензия Apache 2.0 не предусматривает подобных требований, ограничивающих ее применение. В связи с этим в настоящей работе используются сторонние продукты, имеющие только эту лицензию.

## МЕЖДУНАРОДНЫЙ СТАНДАРТ BPMN

Стандарт BPMN — это система условных обозначений, предназначенных для моделирования бизнес-процессов. Каждый моделируемый процесс может быть описан одной или несколькими диаграммами, состоящими из графических элементов, представленных следующими основными категориями:

- объекты потока (flow objects);
- данные (data);
- зоны ответственности (swimlanes);
- соединяющие элементы (connecting objects);
- артефакты (artifacts).

С помощью BPMN можно реализовывать самые различные процессы. Например, на рис. 1 продемонстрирован алгоритм процесса управления выдачей банковских карт.

В данной работе основная задача графического редактора — построение диаграмм с использованием нотации BPMN. Возможности редактора не ограничиваются лишь визуальным представлением процессов — с его помощью также можно

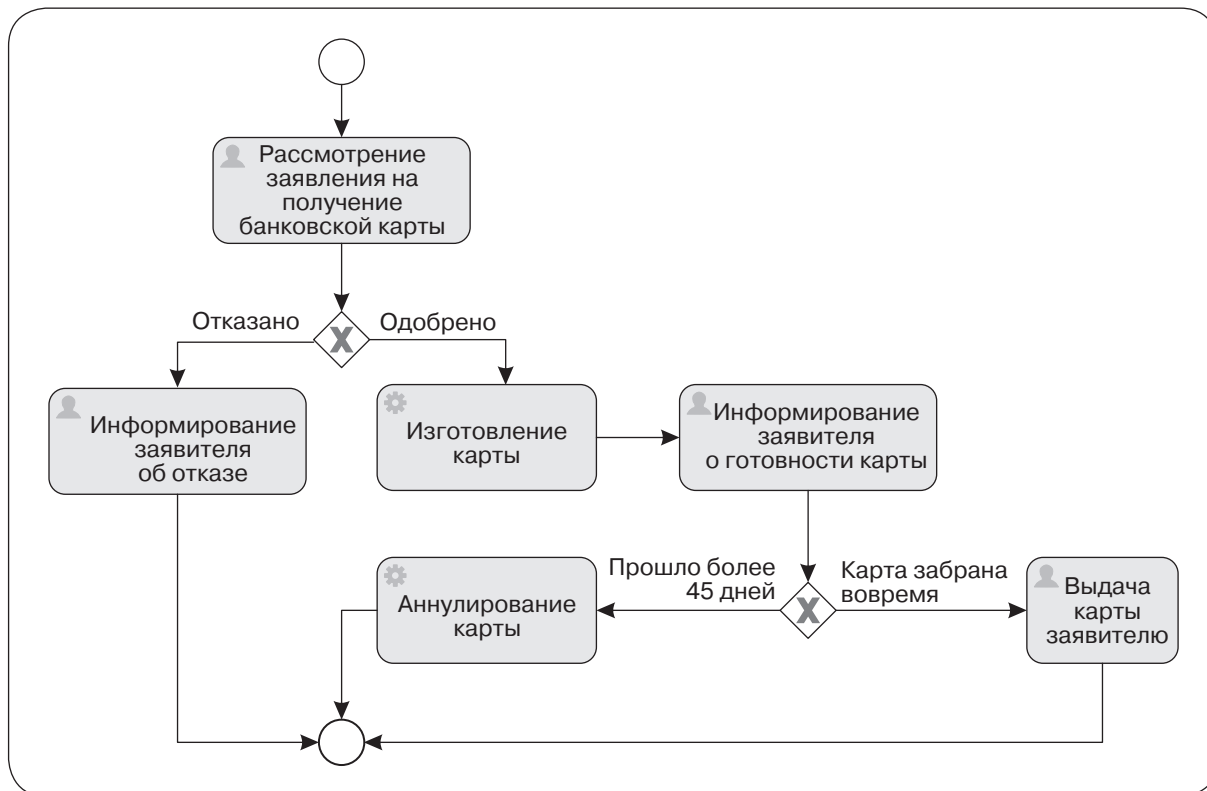


Рис. 1. Пример BPMN-диаграммы (управление выдачей банковских карт)

обеспечить хранение логической модели, описываемой в виде графа.

Полностью реализовать графический редактор вовсе не просто. В связи с тем, что существуют различные варианты готовых решений, целесообразно выбрать подходящую версию существующего редактора и доработать ее.

С учетом особенностей лицензирования мы остановили свой выбор на редакторе Activiti-Designer, который является частью системы Activiti, предназначенной для моделирования бизнес-процессов в виде BPMN-диаграмм и имеющей открытый исходный код (распространяется в рамках лицензии Apache). Activiti обладает не всеми желаемыми и необходимыми функциями, поэтому потребовалось расширение возможностей редактора. Рисунок 2 демонстрирует пример диаграммы, выполненной в редакторе Activiti-Designer.

## ДОРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА

Удобство использования разработанного инструмента особенно важно для непрофессионалов в сфере ИТ.

В графический редактор Activiti нами были добавлены две основные функции: быстрый доступ и видимость элементов.

1. *Быстрый доступ.* Эта функция позволяет отображать на холсте элементы, не перемещая их из панели элементов. Пользователю предоставляется набор комбинаций «CTRL + M» и «CTRL + N» (где M — номер категории, в которой расположен элемент, а N — номер элемента внутри категории). Таким образом, применяя какую-либо комбинацию клавиш из данного набора, соответствующий элемент добавляется на диаграмму в то место, где находится курсор мыши.

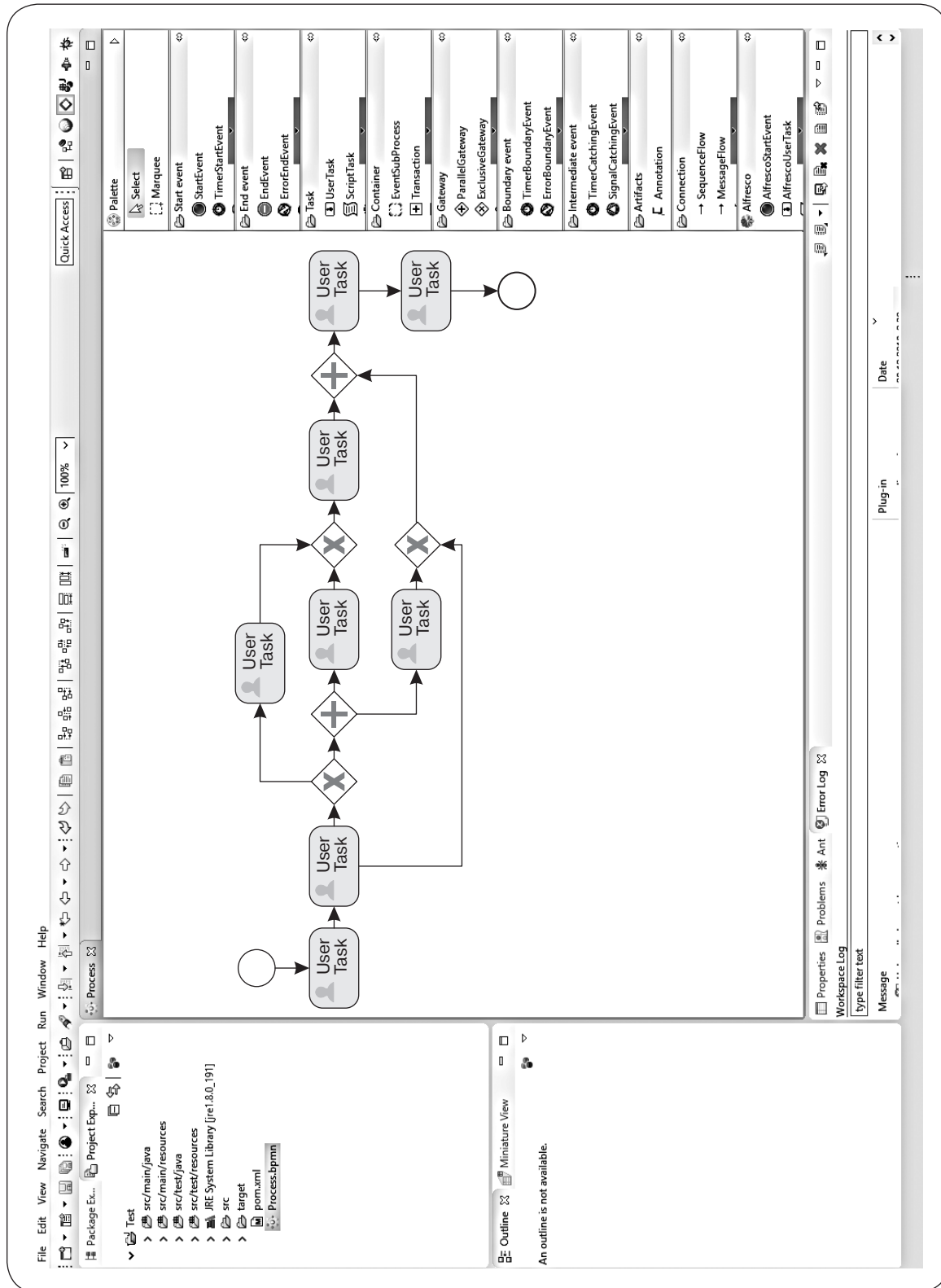


Рис. 2. Пример диаграммы в редакторе Activiti-Designer (скриншот окна Eclipse)

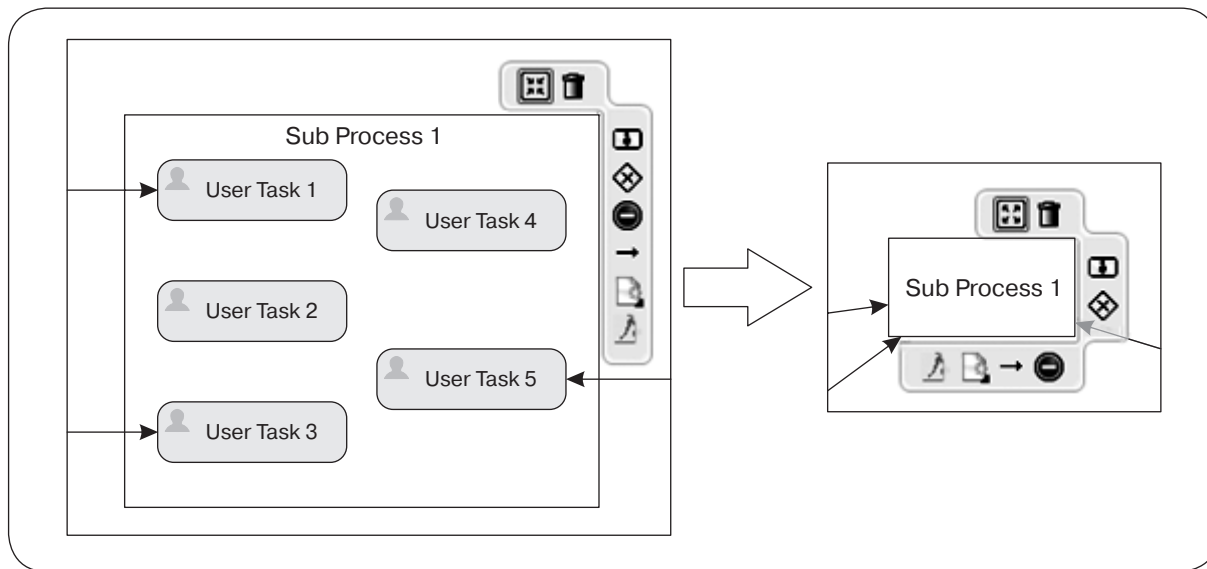


Рис. 3. Пример использования функции «Видимость элементов»

2. *Видимость элементов.* Данная функция также является значимым расширением возможностей редактора. Она позволяет скрывать элементы контейнера в случае громоздких диаграмм и высокого уровня вложенности. Пользователь применяет функции *minimize* и *maximize*, которые добавляются в контекстное меню контейнера в зависимости от того, скрыты или нет его элементы. Функция *minimize* скрывает все элементы контейнера, а последний представляется в виде небольшого прямоугольника с сохранением всех внешних связей, которые присутствовали в его полноразмерном варианте. При применении функции *maximize* контейнер принимает свой изначальный вид.

Пример использования данной функции приведен на рис. 3.

## ПРИМЕРЫ ПРИМЕНЕНИЯ ПЛАТФОРМЫ

Рассмотрим возможные варианты применения платформенного продукта. В настоящее время несколько предприятий вы-

разили заинтересованность в его реализации.

### Управление гибким автоматическим производством

В последние годы на производственных предприятиях становится все больше станков с числовым программным управлением (ЧПУ) [Ловыгин, Теверовский, 2015], что значительно повышает автоматизацию производства и позволяет гарантировать качество продукции. Полная автоматизация таких процессов возможна за счет использования на ряду со станками с ЧПУ складских роботов (робот-тележка, робот-манипулятор и др.), основная задача которых — передача необходимых материалов и инструментов от одного станка к другому. Система, состоящая из таких станков и роботов, образует гибкое автоматическое производство (ГАП).

Как программировать и контролировать работу ГАП? Необходимо не только задать всю последовательность действий, но и описать реакцию на сбойные ситуации, ведь из-за наличия случайного мусора или неровностей поверхности робот

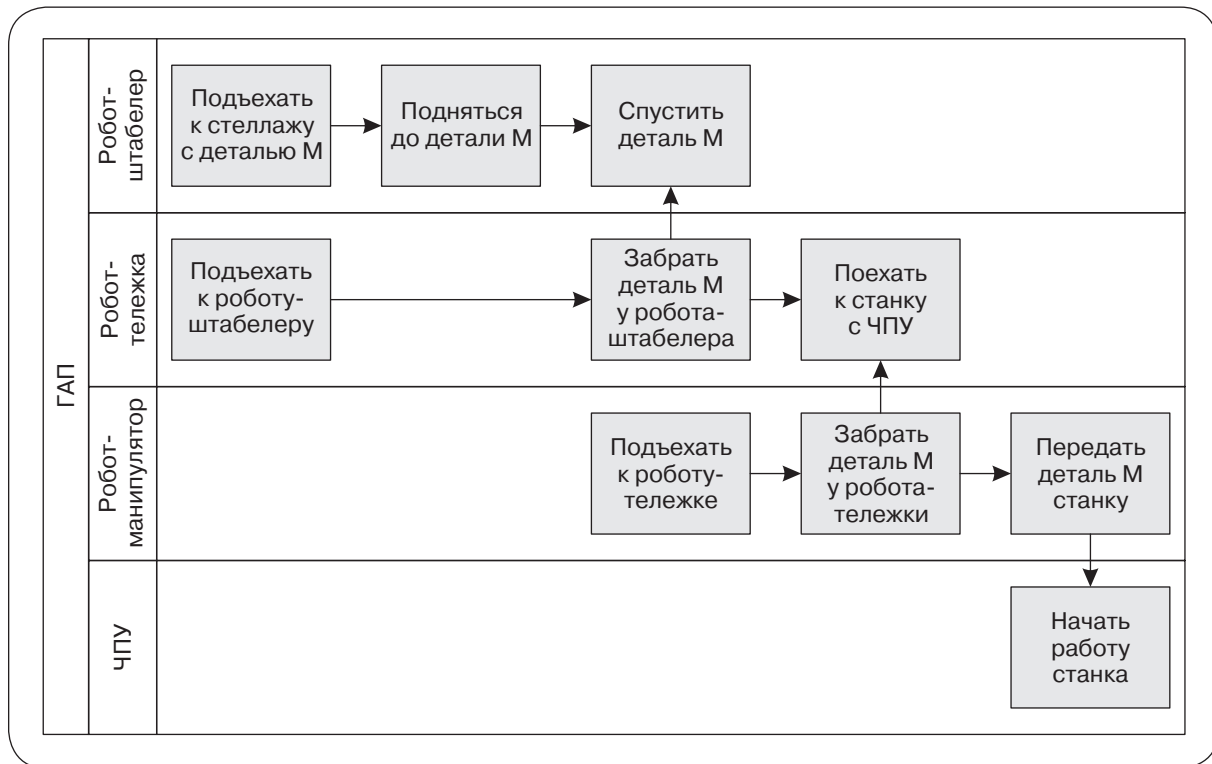


Рис. 4. Пример функционирования системы ГАП в виде BPMN-диаграммы

может сойти с пути или упасть, что способно вызвать нарушение корректной работы системы или даже поломку перевозимого груза.

Функционирование ГАП, включая работу станков с ЧПУ и роботов, может быть описано в виде BPMN-диаграмм (рис. 4). Кроме того, таким способом удобно задавать в том числе и ветки отказов роботов, что позволит быстро реагировать на проблемы и восстанавливать работу производства. Ситуации сбоев в имплементации процесса могут считываться при помощи различных сенсоров и камер [Yousfi et al., 2016].

### Легкий документооборот

В любой компании необходимо проведение отслеживания документов и проверки их статуса. Иными словами, на различных стадиях жизненного цикла конкретного

документа важно знать, где и в каком состоянии он находится.

Данный процесс продемонстрирован в виде BPMN-диаграммы (рис. 5). Диаграмма позволит без особых усилий отслеживать нахождение документа и его текущий статус, так как его движение происходит по построенному графу. Важно учитывать, что во многих компаниях существует система иерархии; при помощи такого представления движения документа можно контролировать, чтобы доступ к нему имели только сотрудники с необходимыми полномочиями.

### Управление поставками и рисками

Проведение анализа поставщиков и связанных с ними рисков — важная задача любой компании, так как сбои поставок могут существенно повлиять на ход производства, например вызвать изменение

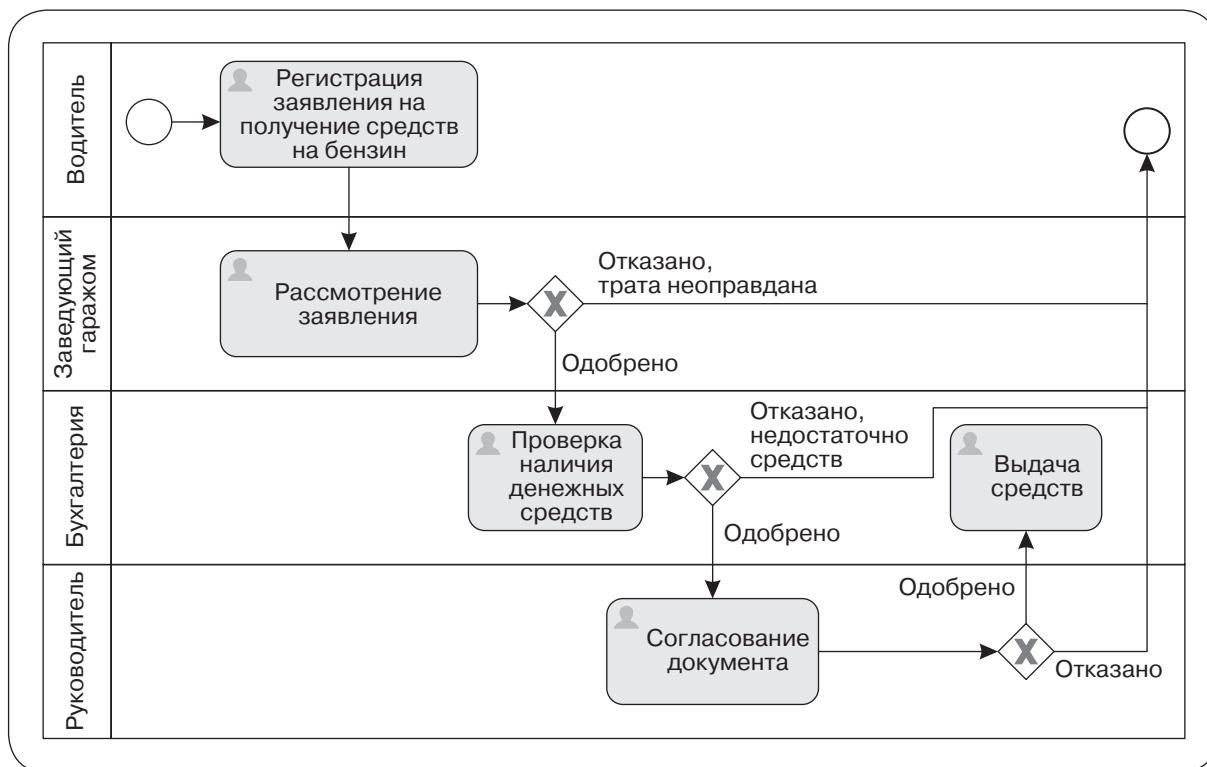


Рис. 5. Пример реализации легкого документооборота в виде BPMN-диаграммы

сроков или увеличение бюджета. Ввиду того что поставщики могут находиться в разных местах производственной цепочки, существует необходимость контролировать всю систему поставок.

Одним из способов описания такой структуры выступает использование BPMN-диаграмм. Оно позволяет внедрить автоматический анализ данных, применить накопленную статистику по каждому поставщику и управлять процессом в целом, что значительно уменьшит пользовательские действия и сократит влияние человеческого фактора.

BPMN-диаграмма обеспечивает важное преимущество в работе, так как с ее помощью можно анализировать работу поставщиков как по срокам выполнения поставок, так и по достигнутым результатам деятельности одновременно. В соответствующей BPMN-диаграмме горизонтально может быть представлена временная шкала,

а вертикально — структурная подчиненность и статус поставок.

## ЛОГИЧЕСКАЯ МОДЕЛЬ

Визуальное (создаваемое пользователем) представление бизнес-процесса принято называть графической моделью, а внутреннее (в виде, например, специального графа), необходимое для генерации отчетов, исполняемых программ и т. д., — логической моделью.

Одна из ключевых задач настоящей статьи — представление логической модели. Удобный и быстрый доступ к данным значительно упрощает последующие действия, связанные с генерацией исполняемой программы, а также разбором и анализом модели [Dos Santos et al., 2019].

Основной причиной хранения информации в виде ориентированных графов



является простота преобразования диаграмм пользователей в графы; кроме того, на них удобно хранить какую-либо дополнительную информацию.

Помимо этого, из-за отсутствия каких-либо ограничений на представление диаграммы возрастает сложность работы с графами. Например, множество циклов и петель мешает определять начальные и конечные вершины описанного процесса, усложняет анализ переходов между элементами, поэтому нами были приняты некоторые ограничения, направленные на повышение структурности моделей.

Учитывая особенности требуемого графа, была выбрана определенная модель его представления. Рассмотрим подробнее полученную структуру.

### Базовый тип

Каждый элемент графа имеет уникальный идентификатор (*id*) и дополнительную информацию об элементе (*info*). Параметр *id* задается при создании элемента и записывается в *info*. В *info* хранятся имя, тип и множество других компонентов элемента. Невозможно не заметить важности этих данных, и именно поэтому *id* и *info* легли в основу базового класса — *BaseItem*.

### Типы данных в графе

Граф состоит из элементов трех типов: *Container* — контейнер, *Node* — узел, *Edge* — ребро. Соответствующие им классы предоставляют всю необходимую информацию об элементах и их связях с другими элементами графа. Следует отметить, что хранение всей информации может быть избыточным и неэффективным, поэтому в качестве связей между элементами используется только *id*.

Важно пояснить, что такое контейнер и для чего он нужен. Контейнер предназначен для хранения любых элементов графа, в нем может лежать сколько угодно

других контейнеров. Кроме того, его можно использовать в качестве вершины.

Каждый элемент диаграммы содержит информацию о родительском контейнере, в котором он находится, причем ребро может иметь два таких контейнера, поскольку его начало и конец могут находиться в каждом из них. Базовым контейнером является холст диаграммы, что означает наличие родительского контейнера для любого элемента графа.

### Представление графа в памяти

За представление графа отвечают отдельные классы — *NodeUtil*, *EdgeUtil* и *ContainerUtil*. Они являются статическими, что позволяет не создавать отдельный экземпляр при каждом применении, а использовать один и тот же для всех обращений. Каждый класс отвечает за актуальный список соответствующих элементов. Модификация классов происходит при любых изменениях диаграммы, например при добавлении, изменении или удалении элемента.

## РЕАЛИЗАЦИЯ ГЕНЕРАТОРА КОДА

Исполняемые программы разных типов производственных процессов существенно различаются по структуре и конечному результату. Например, генератор кода можно реализовать для процессов, соответствующих системе ГАП. Каждый такой процесс представляется в виде диаграммы, на базе которой строится логическая модель с последующей генерацией исполняемого кода (технология CASE — Computer-Aided Software Engineering) [Терехов, 2006].

Основной этап, предшествующий кодогенерации, — анализ и разбор модели. Этот этап необходим для выявления начальных и конечных узлов, циклов, петель и т. д. Полученные на нем данные являются вход-

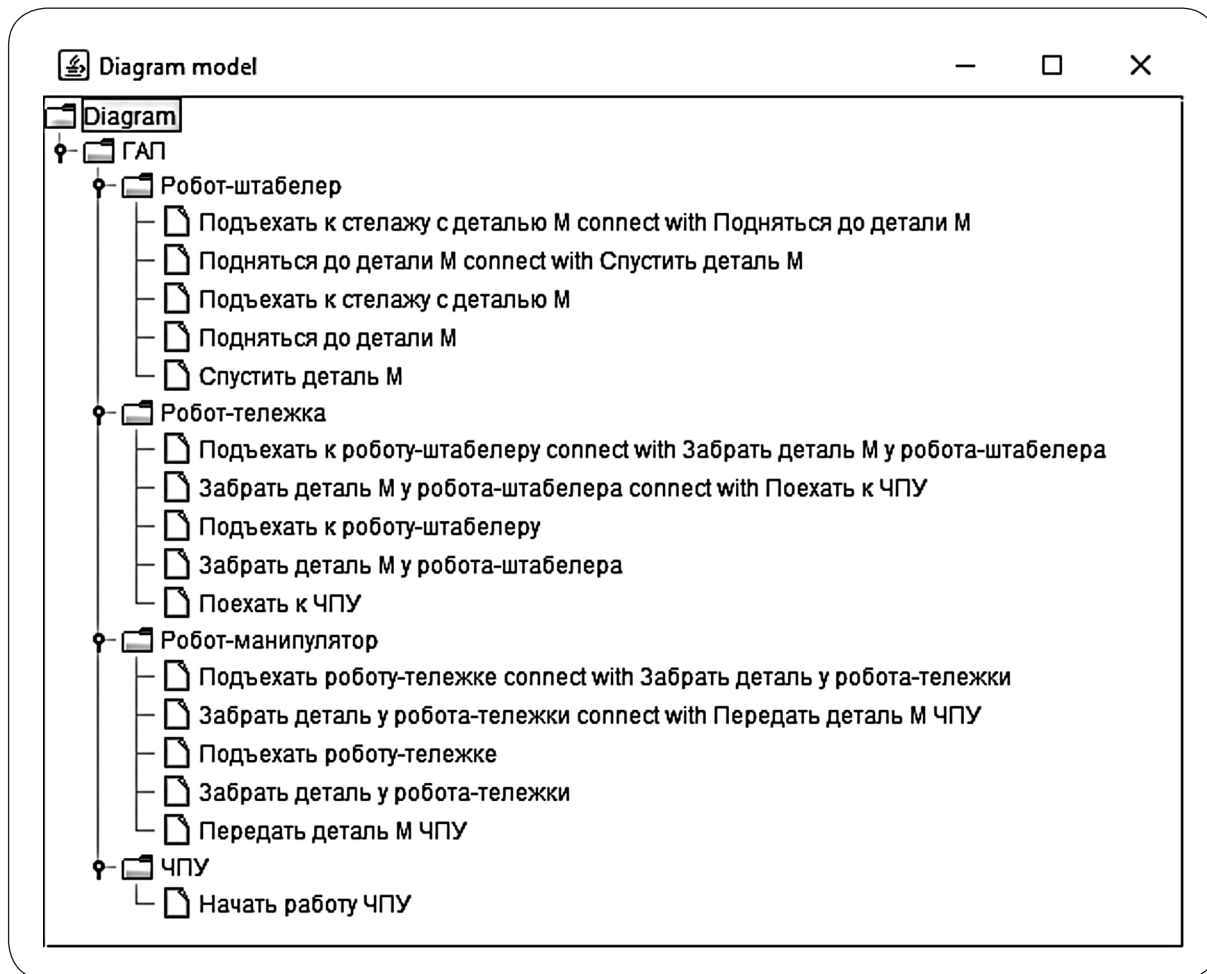


Рис. 6. Пример визуализации модели с помощью функции Show Model

ными для генератора кода, поэтому полезно иметь возможность их визуализировать, благодаря чему обнаруживаются различные неточности и ошибки, упущенные во время построения диаграммы.

Пользователю предоставляется функция Show Model, доступ к которой возможен через основное меню. На рис. 6 приведен пример визуализации модели диаграммы с использованием этой функции.

Применяя данную функцию для какой-либо диаграммы, можно открыть отдельное окно, где все ее элементы отображаются в конкретном виде. Важно отметить, что ребенок узла — это узел, являющийся конечным для какого-либо ребра, вы-

ходящего из него, и контейнер может также выступать в качестве узла, т.е. как иметь детей, так и быть ребенком. Для каждого контейнера, включая базовый, в окне отображаются все соответствующие ему элементы: узлы и их дети. При таком представлении логической модели задача генерации кода становится простой — линейный обход с генерацией нужного фрагмента кода по каждому действию.

## ЗАКЛЮЧЕНИЕ

В то время как изучению АСУ с позиции позитивного подхода посвящено достаточ-

но большое количество исследований (см. напр.: журнал *MIS Quarterly*), в данной работе предложена попытка построения нормативной модели [Шеер, 2000]. В частности, представлен четырехэтапный алгоритм проектирования и моделирования системы управления производственными процессами, а также обсуждается несколько

ко примеров приложений, созданных на его основе. Кроме того, выделяются технологические решения, существенно упрощающие работу с предлагаемыми средствами и обеспечивающие «сквозной» характер ее применения. Описываемая платформа и ее элементы уже нашли практическую реализацию.

## ЛИТЕРАТУРА НА РУССКОМ ЯЗЫКЕ

- Афанасьев А. А. 2014. Технология визуализации данных как инструмент совершенствования процесса поддержки принятия решений. *Инженерный вестник Дона* (4): 32–48.
- Ловыгин А. А., Теверовский Л. В. 2015. *Современный станок с ЧПУ и CAD/CAM система*. М.: ДМК Пресс.
- Лонский И. И. 2015. Информатизация и эволюция общества. *Перспективы Науки и Образования* (2): 29–35.
- Нестеренко Е. А., Козлова А. С. 2018. Направления развития цифровой экономики и цифровых технологий в России. *Экономическая безопасность и качество* 2 (31): 9–14.
- Плотников В. А. 2018. Цифровизация производства: теоретическая сущность и перспективы развития в российской экономике. *Известия Санкт-Петербургского*

*государственного экономического университета* (4): 16–24.

- Полякова Е. В. 2012. Применение способов и методов визуального мышления в современном образовании. *Известия Южного федерального университета. Технические науки* (10): 120–124.
- Терехов А. Н. 2006. *Технология программирования*. М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний.
- Хусяинов Т. М. 2017. Информационная революция и трансформация занятости. *Наука. Мысль* (1–3): 76–79.
- Шеер А. В. 2000. *Моделирование бизнес-процессов*. 2-е изд. Пер. с англ. М.: Весть-Мета Технология. Оригинал: Sheer A.-W. 1999. *ARIS – Business Process Modeling*. Berlin-Heidelberg: Springer-Verlag.

## REFERENCES IN LATIN ALPHABET

- Alotaibi Y. 2017. Graphical business process modelling standards, techniques and languages: a literature review. *International Journal of Business Information Systems* 25 (1): 18–54.
- Dos Santos C. H., Thom L. H., Cota É., Fantinato M. 2019. Supporting BPMN tool developers through meta-algorithms. *International Journal of Business Information Systems* 32 (4): 460–488.
- Vasilecas O., Kalibatiene D., Lavbič D. 2016. Rule- and context-based dynamic business process modelling and simulation. *Journal of Systems and Software* 122: 1–15.

- Yousfi A., Bauer C., Saidi R., Dey A. K. 2016. uBPMN: A BPMN extension for modeling ubiquitous business processes. *Information and Software Technology* 74: 55–68.

### Translation of references in Russian into English

- Afanasief A. A. 2014. A technology for data visualization as an instrument for decision support process enhancement. *Inzhenernyi Vestnik Dona* (4): 32–48. (In Russian)
- Lovygin A. A., Teverovskiy L. V. 2015. *Modern Machine with CNC and CAD/CAM*

- System*. Moscow: DMK Press. (In Russian)
- Lonskiy I. I. 2015. Informatization and evolution of the society. *Perspektivy Nauki i Obrazovaniya* (2): 29–35. (In Russian)
- Nesterenko E. A., Kozlova A. S. 2018. Directions of digital economy and digital technologies development in Russia. *Economicheskaya Bezopasnost i Kachestvo* 2 (31): 9–14. (In Russian)
- Plotnikov V. A. 2018. Digitalization of manufacturing: Theoretical notion and perspectives of development in the Russian economy. *Izvestiya Sankt-Peterburgskogo Gosudarstvennogo Economicheskogo Universiteta* (4): 16–24. (In Russian)
- Polyakova E. V. 2012. Applications of approaches and methods of visual thinking in the contemporary education. *Izvestiya Yuzhnogo Federalnogo Universiteta. Tekhnicheskiye Nauki* (10): 120–124. (In Russian)
- Terekhov A. N. 2006. *Technology of Programming*. Moscow: Internet-university of Information Technologies. (In Russian)
- Husiyanov T. M. 2017. Information revolution and transformation of employment. *Nauka. Mysl* (1–3): 76–79. (In Russian)
- Sheer A.-W. 2000. *Business Process Modeling*. 2-nd ed. Russian translation. Moscow: Vest-Meta Tehnologiya. (In Russian)

*Статья поступила в редакцию  
1 ноября 2019 г.  
Принята к публикации  
27 декабря 2019 г.*

---

### ***Business Process Modeling in the Digital Age***

***A. N. Terekhov, M. V. Platonova***

Mathematics and Mechanics Faculty, St. Petersburg State University, Russia

Newly emerged in the Russian society trend towards digitalization is gaining increasing number of applications. Business processes modeling based on modern IT approaches has a large potential for improving the efficiency and reliability of various manufacturing systems. Unfortunately, studies that offer specific tools for formal description of manufacturing processes are rare. In this paper we propose a four-stage system for modeling business processes: design, visualization, construction of a logical model and, finally, generation of a computer code (CASE technology). This technology for business processes modeling, as well as specific products based on it, can be widely used in manufacturing processes, from document workflow automation to the operation of numerically controlled machines. The proposed tool can help to increase the productivity of personnel of organizations and optimize manufacturing processes. The automatic code generation eliminates one of the main drawbacks of software development — the code does not contain input errors.

***Keywords:*** digitalization, visualization, business process, workflow, computer numerical control, BPMN, CASE technology.

***JEL:*** M53.

***For citation:*** Terekhov A. N., Platonova M. V. 2019. Business process modeling in the digital age. *Russian Management Journal* 17 (4): 487–498.

*Initial Submission: November 01, 2019  
Final Version Accepted: December 27, 2019*